7

Unicode und die richtigen Sprachparameter

Computer können die unterschiedlichsten Zeichen darstellen, aber nicht „verstehen“ (verarbeiten). Konkret bedeutet das, dass jedes Schriftzeichen, das auf dem Bildschirm oder in einem Papierausdruck erscheint, nur ein Bild ist, dem intern eine Zahl zugeordnet ist. Eine Sammlung solcher Schriftzeichen und der zugehörigen Zahlen nennt man einen *Zeichensatz.*

Ursprünglich unterstützten Computer nur einen begrenzten Zeichenvorrat, PCs z. B. arbeiteten nur mit dem *ASCII-Zeichensatz (American Standard Code for Information Interchange),* der US-Fassung des ISO-7-Bit-Codes (auch *ISO 646*). Er umfasste das englische Alphabet, die Ziffern 0 bis 9 und einige Sonder- und Interpunktionszeichen. Erste lokale Anpassungen ergaben sich durch nationale Varianten des ISO-7-Bit-Codes, in denen 12 der ASCII-Sonderzeichen durch länderspezifische Zeichen wie Umlaute oder das skandinavische Å ersetzt wurden. Mit dem auf 8 Bit *erweiterten IBM-Zeichensatz* gesellten sich einige griechische und mathematische Zeichen hinzu und erst der Zeichensatz *ISO-8859-1* versorgte den Anwender mit allen für westeuropäische Sprachen erforderlichen Sonderzeichen wie Umlaute, diverse Akzentzeichen usw. Entsprechend gab es für andere Sprachumgebungen eigene Zeichensätze, etwa *KOI8-R* für das russische Alphabet oder *JIS X 0208* für japanische *Ideogramme* (Schriftzeichen, die jeweils für einen ganzen Begriff stehen).

In jüngster Zeit jedoch entstand immer mehr der Bedarf, Computerumgebungen international zu gestalten. Daher wurde ein universeller Zeichensatz geschaffen – *Unicode.* Dieser Zeichensatz deckt – angefangen bei deutschen Umlauten, speziellen Interpunktionszeichen oder Ligaturen über japanische Schriftzeichen bis hin zu indianischen Zeichen oder Runen – ganz unterschiedliche Schriftzeichen ab und ist identisch mit dem Zeichensatz *ISO/IEC 10646.* Im Unicode-Zeichensatz lassen sich theoretisch 1 114 111 Schriftzeichen codieren, das ist mehr als ausreichend für alle Sprachen der Welt.

Zeichensätze in XML

Der XML-Zeichensatz ist Unicode. Das heißt, ein XML-Parser (oder -Prozessor) muss jedes Unicode-Zeichen unterstützen. Das heißt aber nicht, dass jedes XML-Dokument in Unicode gespeichert werden muss, denn die meisten XML-Parser unterstützen zusätzlich die wichtigsten anderen Zeichensätze wie *ISO-8859-1* (für westeuropäische Zeichen) oder weitere Zeichensätze nach dem ISO-Standard. Daher ist es möglich, auch in diesen Formaten gespeicherte XML-Dokumente zu verwenden. Einzige Voraussetzung: Der Fremdzeichensatz muss in der *XML-Deklaration* des Dokuments deklariert sein. Anhand dieser Information kennt der Parser die Codierung des Ausgangstexts und konvertiert ihn in Unicode. Wenn Sie nun in einem solchen XML-Dokument mit eingeschränktem Zeichenvorrat dennoch ein besonderes Zeichen benötigen (etwa weil Sie ein Zitat in japanischen Schriftzeichen anführen möchten oder ganz einfach ein Copyright-Zeichen benötigen), so ist auch das möglich und zwar in Form einer *Zeichenreferenz*. Der Zahlencode in Zeichenreferenzen bezieht sich immer auf den Unicode-Zeichensatz und nicht etwa auf den aktuell deklarierten.

Lesen Sie nach ...

- Wie Sie den aktuellen Zeichensatz deklarieren, erfahren Sie in Kapitel 6 im Abschnitt *Die XML-Deklaration*.
- Zeichenreferenzen werden ebenfalls in Kapitel 6 erläutert.

Konvertierungs-Tools

Wenn Sie mit den Texteditoren auf Ihrem System nicht zufrieden sind, können Sie sich einen der folgenden Editoren und Konvertierungs-Tools vom Internet herunterladen, die alle Unicode als Dateiformat unterstützen:

TOOL	DOWNLOAD-SITE
EmEditor	*http://www.emurasoft.com/emeditor3/index.htm*
recode	*http://www.iro.umontreal.ca/contrib/recode/*
native2ascii	*http://java.sun.com/j2se/1.3/docs/tooldocs/solaris/native2ascii.html* (Solaris-Version) *http://java.sun.com/j2se/1.3/docs/tooldocs/win32/native2ascii.html* (Windows-Version)

In welchem Zeichensatz ist ein Dokument gespeichert?

Ein XML-Dokument speichern Sie entweder als reine Textdatei oder in einem Unicode-Format. Doch in welchem Zeichensatz bzw. in welcher Codierungsform Ihr Dokument genau gespeichert ist, ist dennoch ungewiss.

Wenn Sie einen Unicode-Editor haben, sind Sie fein raus, denn der Parser kann an der ersten Bytefolge der Datei selbst ermitteln, welche Codierungsform aktiv ist: Beginnt die Datei mit FE FF, ist es *UTF-16BE* (Big Endian), mit FF FE ist es *UTF-16LE* (Little Endian), in allen übrigen Fällen wird *UTF-8* angenommen.

Bei einer normalen Textdatei richten sich die Texteditoren zunächst nach der Ländereinstellung Ihres Systems (kann unter *Windows* etwa in der Systemsteuerung geändert werden) und nach dem Tastaturtreiber (unter Windows mit *Systemsteuerung/Tastatur/Sprache* einstellbar). Für die deutsche Sprache wird in der Regel ein spezieller, für westeuropäische Sprachen geeigneter Zeichensatz verwendet. Auf Unix-Systemen ist dies meist der Zeichensatz *ISO-8859-1.* Das ist prima, denn dieser Zeichensatz wird von XML-Parsern im Allgemeinen unterstützt. Auf Windows-Systemen jedoch wird der Microsoft-eigene Zeichensatz *Cp1252* (auch *Windows-1252*) verwendet, der zwar fast identisch ist mit *ISO-8859-1* und sich häufig als dieser ausgeben lässt, doch sind die Codepositionen einiger Steuerzeichen mit Textzeichen gefüllt, was auf manchen Plattformen fatale Folgen haben kann, wenn diese Zeichen im XML-Dokument vorkommen. Auf Macintosh-Systemen wird ebenfalls ein proprietärer Zeichensatz verwendet, *MacRoman.* Dieser umfasst zwar als Teilmenge den ASCII-Zeichensatz und könnte daher für englische Texte als *ISO-8859-1* oder *UTF-8* durchgehen, für alle anderen Sprachen sollten diese Dateien jedoch konvertiert werden. Für die ISO-Zeichensätze anderer Sprachgebiete gibt es ähnliche Mac- und Windows-Entsprechungen, von denen für XML-Dokumente gleichermaßen abzuraten ist.

Welcher Zeichensatz für welche Sprache?

In der nachfolgenden Liste können Sie nachschlagen, welcher Zeichensatz sich für eine bestimmte Sprache eignet. Unicode, der XML-Standardzeichensatz, umfasst alle Sprachen (siehe den Abschnitt *Unicode*).

SPRACHE	ISO-ZEICHENSATZ	ANDERE
Afrikaans (af)	iso-8859-1	windows-1252
Albanisch (sq)	iso-8859-1	windows-1252
Arabisch (ar)	iso-8859-6	
Baskisch (eu)	iso-8859-1	windows-1252
Belorussisch (be)	iso-8859-5	
Bulgarisch (bg)	iso-8859-5	
Dänisch (da)	iso-8859-1	windows-1252
Deutsch (de)	iso-8859-1	windows-1252
Englisch (en)	iso-8859-1	windows-1252
Esperanto (eo)	iso-8859-3*	
Estnisch (et)	iso-8859-15	
Färöisch (fo)	iso-8859-1	windows-1252
Finnisch (fi)	iso-8859-1	windows-1252
Französisch (fr)	iso-8859-1	windows-1252
Galicisch (gl)	iso-8859-1	windows-1252
Griechisch (el)	iso-8859-7	
Hebräisch (iw)	iso-8859-8	
Holländisch (nl)	iso-8859-1	windows-1252
Inuit-Sprachen (Eskimo-Sprachen)	iso-8859-10*	
Irisch (ga)	iso-8859-1	windows-1252
Isländisch (is)	iso-8859-1	windows-1252
Italienisch (it)	iso-8859-1	windows-1252
Japanisch (ja)	shift_jis	iso-2022-jp
Katalanisch (ca)	iso-8859-1	windows-1252
Kroatisch (hr)	iso-8859-2	windows-1250
Lappisch (lap)	iso-8859-10***	
Lettisch (lv)	iso-8859-13	windows-1257
Litauisch (lt)	iso-8859-13	windows-1257
Maltesisch (mt)	iso-8859-3*	
Mazedonisch (mk)	iso-8859-5	windows-1251
Norwegisch (no)	iso-8859-1	windows-1252
Polnisch (pl)	iso-8859-2	
Portugiesisch (pt)	iso-8859-1	windows-1252
Rumänisch (ro)	iso-8859-2	
Russisch (ru)	iso-8859-5	koi8-r
Schottisch (gd)	iso-8859-1	windows-1252
Schedisch (sv)	iso-8859-1	windows-1252
Serbisch-Kyrillisch (sr)	iso-8859-5***	windows-1251
Serbisch-Lateinisch (sr)	iso-8859-2	windows-1250
Slovakisch (sk)	iso-8859-2	
Slovenisch (sl)	iso-8859-2	windows-1250
Spanisch (es)	iso-8859-1	windows 1252
Thai	iso-8859-11	
Tschechisch (cs)	iso-8859-2	
Türkisch (tr)	iso-8859-9	windows-1254
Ukrainisch (uk)	iso-8859-5	
Ungarisch (hu)	iso-8859-2	

* wird nur von wenigen Browsern unterstützt

** für Lappisch gibt es keinen 2-stelligen Ländercode, in NISO Z39.53 wurde ein 3-stelliger (lap) vorgeschlagen

*** Serbisch lässt sich sowohl in lateinischer (häufiger) wie auch in kyrillischer (meist windows-1251) Schrift schreiben

ZEICHENSATZ	SCHRIFT	SPRACHRAUM (SPRACHEN)
ISO-8859-1 (*Latin 1*, 1987)	Lateinisch	Westeuropäisch (af, ca, da, *de*, en, es, eu, fi, fo, fr, ga, gd, gl, is, it, nl, no, pt, sq, sv)
ISO-8859-2 (*Latin 2*, 1987)	Lateinisch	Zentral- und Osteuropäisch (cs, *de*, en, hr, hu, pl, ro, sk, sl, sr)
ISO-8859-3 (*Latin 3*, 1988)	Lateinisch	Esperanto, Maltesisch und Türkisch (eo, mt, tr)
ISO-8859-4 (*Latin 4*, 1988)	Lateinisch	Baltisch (et, lap, lt, lv) Neuer: *ISO-8859-10* oder *ISO-8859-13*
ISO-8859-5 (1988)	Lateinisch + Kyrillisch	Slavisch (bg, be, mk, ru, uk, sr)
ISO-8859-6 (1987)	Lateinisch + Arabisch	Arabisch ohne Farsi und Urdu (ar)
ISO-8859-7 (1987)	Lateinisch + Griechisch	Neugriechisch (el)
ISO-8859-8 (1988)	Lateinisch + Hebräisch	Hebräisch und Jiddisch (iw)
ISO-8859-9 (*Latin 5*, 1990)	Lateinisch	Westeuropäisch (ohne Isländisch) und Türkisch (af, ca, da, *de*, en, es, eu, fi, fo, fr, ga, gd, gl, it, nl, no, pt, sq, sv, tr)
ISO-8859-10 (Latin 6)	Lateinisch	Baltisch (et, is, lap, lt, lv)
ISO-8859-11	Lateinisch + Thai	Thai
ISO-8859-13 (Latin 7)	Lateinisch	Baltisch (et, is, lap, lt, lv)
ISO-8859-14 (Latin 8)	Lateinisch	Keltisch (ga, gd, Walisisch)
ISO-8859-15 (Latin 9, Latin 0)	Lateinisch	Westeuropäisch (mit Eurozeichen) (af, ca, da, *de*, en, es, eu, fi, fo, fr, ga, gd, gl, is, it, nl, no, pt, sq, sv)

Wichtige ISO-Zeichensätze

In der dieser Tabelle sind die ISO-8859-Zeichensätze sowie die zugehörigen Schriften und Sprachen aufgelistet. Jeder dieser Zeichensätze umfasst als Basis den ASCII-Zeichensatz, d. h., die ersten 128 Zeichen sind identisch codiert.

Lesen Sie weiter ...

In Anhang B finden Sie eine Tabelle allen Zeichen des Zeichensatzes *ISO 8859-1* sowie deren Entsprechung im Unicode-Zeichensatz.

Unicode

Wie bereits erwähnt, ist Unicode der XML-Zeichensatz, also der Zeichensatz, der von einem XML-Parser mindestens unterstützt werden muss. Er ist identisch mit dem Zeichensatz *ISO/IEC 10646* und umfasst die Schriftzeichen der meisten bekannten – lebenden wie auch toten – Sprachen. Die Unicode-Website finden Sie unter *http://www.unicode.org*.

Klasse! Endlich keine Probleme mehr in Sachen Internationalisierung – ganz so einfach ist es leider nicht. Denn auf Ihrem System werden standardmäßig nur die Unicode-Teilmengen installiert, die für Ihren Sprachraum relevant sind. Dasselbe gilt für die Tastaturtreiber und Bildschirmschriften. Aber immerhin haben Sie über die *Zeichenreferenzen* die Möglichkeit, jedes Unicode-Zeichen zu codieren, und wenn das Zielsystem den betreffenden Sprachraum abdeckt, kann es auch dargestellt werden.

Lesen Sie weiter ...

Die Codierungsformen und -schemata des Unicode-Zeichensatzes werden im Abschnitt *Das Zeichencodierungsmodell* erläutert.

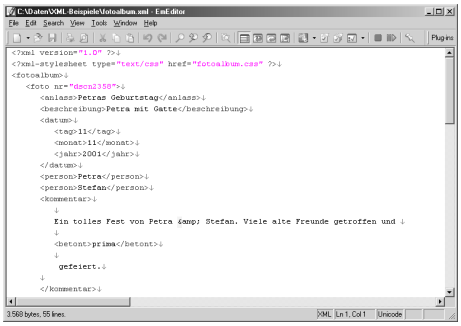

Abbildung 7.1: *EmEditor* kennzeichnet die XML-Auszeichnungen farblich ...

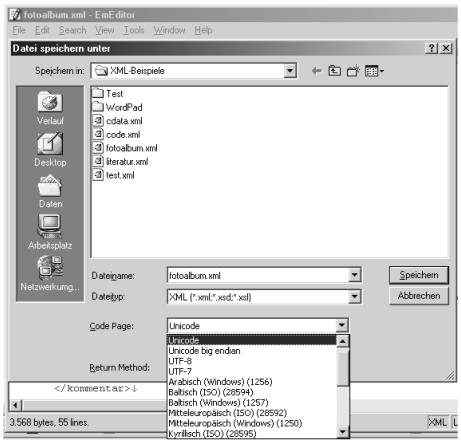

Abbildung 7.2: ... und verfügt über eine Vielzahl verschiedener Codierungsschemata (Code Pages).

Fit für Unicode?

Zeit für ein Browser-Review:

- *Amaya 6* kann eine XML-Datei, nachdem sie in Unicode gespeichert worden ist, leider gar nicht darstellen.
- *Netscape 7* und *Mozilla 1* zeigen nur noch den Text an. Den allerdings richtig und mit Umlauten.
- *Internet Explorer 6* und *Opera 6* haben keinerlei Probleme mit der Unicode-Interpretation. So soll es sein!

Unicode-Editoren

In *WordPad* können Sie ein Dokument im Format *Unicode-Textdokument* speichern, es wird dann die Unicode-Codierungsform *UTF-16* verwendet.

Winword 97 verfügt bereits über den Dateityp **UNICODE-TEXT,** auch hier wird die Datei in der Codierungsform *UTF-16* gespeichert.

Seit *Winword 2000* können Sie das Dateiformat **CODIERTER TEXT** verwenden, Sie erhalten dann eine Auswahl an zahlreichen Zeichensätzen, in denen sich das Dokument speichern lässt: von Arabisch über Baltisch, Türkisch und Unicode bis hin zu schlichtem Westeuropäisch. An Unicode-Codierungsformen wird angeboten: **UNICODE** (= UTF-16, Little-Endian), **UNICODE (BIG-ENDIAN)** (= UTF-16, Big-Endian), **UNICODE (UTF-7)** und **UNICODE (UTF-8).** Wichtig sind **UNICODE** und **UNICODE (UTF-8).**

Seit Windows XP können Sie auch im *Editor (Notepad)* Ihre Dateien in Unicode-Formaten speichern.

EmEditor schließlich ist ein feines Shareware-Tool, das verschiedene Unicode-Codierungsschemata unterstützt. Ein „Schmankerl“ am Rande: Die XML-Auszeichnungen werden in diesem Editor automatisch farblich gekennzeichnet, wodurch er sich generell gut als XML-Editor eignet. Den *EmEditor* finden Sie im Internet unter *http://www.emurasoft.com/emeditor3/index.htm*.

Unicode-Schriften

Zwar haben Sie mit Unicode alle Schriftzeichen der Welt zur Verfügung; allerdings nur, wenn auf Ihrem System auch die richtigen Schriften (Fonts) installiert sind. Wenn Sie bestimmte Unicode-Schriftzeichen anzeigen wollen, auf Ihrem System aber die entsprechenden Schriften fehlen, können Sie einige Unicode-Schriften vom Internet herunterladen.

Arial Unicode MS

Microsoft stellt eine Unicode-Schrift zur Verfügung, die alle im Unicode-2.1-Standard definierten Schriftzeichen umfasst (ca. 40 000). Sie finden sie unter der Adresse *http://office.microsoft.com/downloads/2000/aruniupd.aspx*. Diese Schrift wird zwar für *Microsoft Publisher 2000* angeboten, ist aber nach der Installation wie jede andere Schrift im gesamten System verfügbar.

Lucida Sans Unicode

Eine weitere, relativ umfangreiche Unicode-Schrift, *Lucida Sans Unicode,* finden Sie auf der Site *http://www.hegel.de/allghinweise.html* zum Download. Sie umfasst Ostmitteleuropäisch, Kyrillisch, Baltisch, Neugriechisch, Türkisch und Hebräisch.

Bitstream Cyberbit

Auf der Website *ftp://ftp.netscape.com/pub/communicator/extras/fonts/windows/* schließlich finden Sie die Unicode-Schrift *Bitstream Cyberbit* der Firma Bitstream sowie zwei Teilmengen dieser Schrift. Sie sind für einen einzelnen Benutzer frei verfügbar.

Abbildung 7.3: Mit der richtigen Schrift auf Ihrem System lässt sich die offizielle Seite zum FIFA Weltpokal 2002 auch im Original anzeigen.

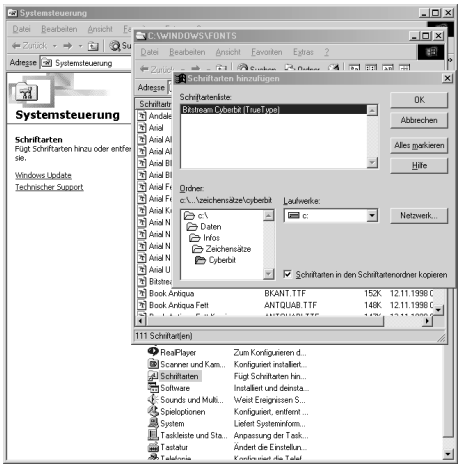

Abbildung 7.4: Auf Windows-Systemen werden Schriften mithilfe des Dienstprogramms SCHRIFTARTEN installiert.

✔ Tipps

- Mithilfe des Font-Servers *xfsft (http://www.dcs.ed.ac.uk/home/jec/programs/xfsft/)* lassen sich True-Type-Schriften (TTF) auch auf Unix-Systemen einsetzen.
- Weitere frei erhältliche Schriften finden Sie unter den folgenden URLs:

  http://www.top20free.com/

  http://www.freepcfonts.com/index.html

  http://www.freewarefonts.com/index.html

Drei ZIP-Pakete stehen zur Auswahl:

- *Cyberbit.ZIP*

  Die ursprüngliche mehrsprachige *Bitstream Cyberbit.* Diese Schrift umfasst nahezu 30 000 Zeichen, u. a. Ostmitteleuropäisch, Kyrillisch, Baltisch, Neugriechisch, Türkisch, Hebräisch, Arabisch, Chinesisch, Koreanisch und Japanisch.

- *CyberCJK.ZIP*

  Eine Teilmenge der *Bitstream Cyberbit.* Sie umfasst die CJK-Schriftzeichen, also Chinesisch, Japanisch und Koreanisch. (Hier ist jedoch noch kein Eurozeichen enthalten.)

- *CyberBas.ZIP*

  Eine weitere Teilmenge. Sie umfasst *Cyberbit* minus *CyberCJK.* Diese Schrift eignet sich insbesondere für ISO-8859-x-Sprachen.

So installieren Sie eine Schrift (TTF-Datei):

1. Öffnen Sie das heruntergeladene ZIP-Archiv mit einem Windows-Programm wie *WinZip.* Es umfasst eine der drei folgenden Schriftdateien: *Cyberbit.ttf, CyberCJK.ttf* oder *Cyberbas.ttf.*
2. Kopieren Sie die gewünschte(n) Schriftdatei(en) in einen geeigneten Ordner.
3. Öffnen Sie die Windows-Systemsteuerung und darin das Dienstprogramm **SCHRIFTARTEN.**
4. Wählen Sie **DATEI/NEUE SCHRIFTART INSTALLIEREN.**
5. Wählen Sie im Dialogfenster **SCHRIFTARTEN HINZUFÜGEN** den Ordner aus, in dem Sie die Schriftdatei abgelegt haben.
6. Markieren Sie im Feld **SCHRIFTARTENLISTE** die Schrift **BITSTREAM CYBERBIT (TRUE TYPE)** und wählen Sie **OK.**

   Sie können die Schriftdatei auch einfach in den Ordner *C:\WINDOWS\FONTS* kopieren, damit die Schrift installiert wird.

Der UniBook Character Browser

Wenn Sie häufig XML-Dokumente in verschiedenen Sprachen erstellen und verarbeiten, sollten Sie sich unbedingt den *UniBook Character Browser* besorgen. Ein sehr nützliches, relativ intuitives Tool, mit dem Sie das Gesamtpaket an Unicode-Zeichentabellen schnell im Griff haben. Allerdings sind auch in UniBook nur die Zeichen sichtbar, die gemäß Ihrer Ländereinstellungen auf Ihrem System unterstützt werden. *UniBook 3.0* können Sie sich von der Seite *http://www.unicode.org/unibook* herunterladen.

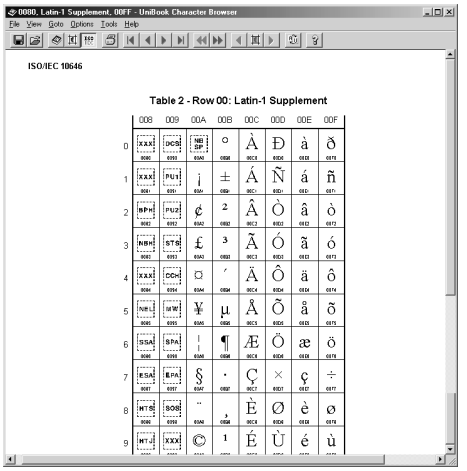

Abbildung 7.5: Der *UniBook Character Browser* mit der Zeichentabelle *Latin-1 Supplement* (in der die Sonderzeichen der deutschen Sprache zu finden sind).

So installieren Sie UniBook:

1. Kopieren Sie alle Dateien des ZIP-Archivs in ein Verzeichnis, etwa *...\Programme\UniBook*.
2. Öffnen Sie die Datei *install.bat*.
3. Bestätigen Sie die diversen Dialogfelder.

Das Programm wird geöffnet und im Programmverzeichnis wird die Datei *Unibook.exe* angelegt. Um das Programm später auszuführen, öffnen Sie diese Datei.

Einige Tipps zur Anwendung von UniBook:

- Mit den Symbolen können dene Anzeigeformate auswählen; ich finde die ISO-Anzeige am übersichtlichsten.
- Mit Strg + B wählen Sie eine Zeichentabelle nach ihrem Namen aus.
- Jede Tabelle wird zweimal angezeigt, zunächst eine Tabelle, in der jedes Zeichen in einem eigenen Karo angezeigt wird, dann eine Tabelle, in der eine Beschreibung der einzelnen Zeichen aufgelistet ist. Mit Bild ↓ bzw. Bild ↑ schlagen Sie die nächste bzw. vorige Tabelle auf.
- Um einen anderen Tabellenausschnitt anzuzeigen, verwenden Sie die Bildlaufleiste oder die Tasten ↓ bzw. ↑.
- Im Menü **VIEW** können Sie die Ansicht vergrößern.
- Per Klick auf die einzelnen Zeichen lässt sich eine Beschreibung einblenden.
- Vorsicht! Der Aufruf des Suchfensters mithilfe von Strg + F führt mitunter zu einem Programmabsturz, ansonsten ist das Programm aber stabil.

Das Zeichencodierungsmodell

Wie bereits erwähnt, ist für den Computer ein Text nur eine Folge von Zahlen, genauer gesagt von zwei Zahlen: 0 und 1. Nun wird eine Sammlung von Schriftzeichen – ein so genanntes *Zeichenrepertoire* (auch *Zeichenvorrat*) – in mehreren Stufen codiert, bis eine vom Computer lesbare Form erreicht ist. Dieser Prozess ist im *Zeichencodierungsmodell* umschrieben.

Zwar ist eine genaue Kenntnis des Zeichencodierungsmodells für die Arbeit mit XML nicht erforderlich, doch gibt es in diesem Zusammenhang eine Vielzahl von Konzepten und Begriffen, die immer wieder durcheinander geraten, daher sollen hier die Grundzüge des Modells kurz vorgestellt werden.

Bis ein Zeichen auf dem Computer gespeichert und wieder korrekt abgerufen werden kann, werden – ausgehend vom *abstrakten Zeichenrepertoire* – drei Codierungsstufen durchschritten:

1. Der codierte Zeichensatz (CCS, Coded Character Set)
2. Die Codierungsform (CEF, Character Encoding Form)
3. Das Codierungsschema (CES, Character Encoding Scheme)

Das abstrakte Zeichenrepertoire

Ein abstraktes Zeichenrepertoire ist die Sammlung aller abstrakter Zeichen, die zu codieren sind, also ein Alphabet oder ein Satz an Symbolen. Es gibt *geschlossene Repertoires,* die sich nicht erweitern lassen (etwa das Repertoire eines ISO-8859-Zeichensatzes) sowie *offene* (etwa der Zeichenvorrat des Unicode-Zeichensatzes, der ja universell ist, und somit potenziell jedes abstrakte Zeichen umfasst, das jemals codiert werden soll). Auch die Repertoires der Windows-Zeichensätze sind offen, denn wenn ein neues Zeichen hinzukommt (wie jüngst das Eurozeichen), so wird es einfach ergänzt, während die

ISO-Reihe einen gänzlich neuen Zeichensatz bildet (z. B. *ISO-8859-15*). Damit abstrakte Zeichen eindeutig identifizierbar sind, hat man für sie feste Bezeichnungen in englischer Sprache eingeführt, so dass etwa *LATIN CAPITAL LETTER A WITH DIARESSIS* immer den Buchstaben *Ä* bezeichnet, und zwar unabhäng davon, in welchem Zeichensatz er vorkommt und mit welcher Codenummer er dort codiert ist.

Der codierte Zeichensatz

In einem codierten Zeichensatz ist ein abstraktes Zeichenrepertoire einem Satz an ganzen Zahlen zugeordnet, wobei dieser Zahlenbereich nicht fortlaufend sein muss. Ein so zugeordnetes (oder *codiertes*) abstraktes Zeichen ist dann Teil des betreffenden Zeichensatzes. Die zugeordneten Zahlen werden auch *Codenummern, Codepositionen, Codepunkte* oder *Skalarwerte* genannt. Diesen Wert geben Sie (bezogen auf den Unicode-Zeichensatz) an, wenn Sie ein Zeichen in einem XML-Dokument mithilfe einer *Zeichenreferenz* benennen. Die Codeposition kann in dezimaler oder hexadezimaler Schreibweise angegeben werden. Beispiele für codierte Zeichensätze sind *ISO-8859-1, Unicode, ISO/IEC-10646.*

Andere Bezeichnungen für einen codierten Zeichensatz sind *Zeichencodierung, codiertes Zeichenrepertoire, Zeichensatzdefinition, Codepage, Character-set, Character-encoding* (und unter Windows gelegentlich auch *Script*).

Die Darstellung der Codenummern

Die Codenummern eines Zeichensatzes werden immer wieder anders angegeben, nämlich in verschiedenen Zahlensystemen. Die Tabelle unten verdeutlicht dies anhand von 3 Beispielen: Dem Buchstaben A, dem Eurozeichen und dem hebräischen Zeichen ℵ.

Abstraktes Zeichen	Codenummer im Unicode-Zeichensatz		
	Dezimalschreibweise	Hexadezimalschreibweise	Binäre Schreibweise
A	65	00 41	0100 0001
€	8 364	20 AC	0010 0000 1010 1100
ℵ	1 488	05 D0	0101 1101 0000

Lesen Sie nach ...

Zeichenreferenzen werden in Kapitel 6, Abschnitt *Zeichenreferenzen* erläutert.

Lesen Sie weiter ...

Die Codepositionen der westeuropäsichen Zeichen finden Sie in der Tabelle in Anhang B.

Byte für Byte

Die folgende Tabelle zeigt auf, wie dieselbe Binärzahl in verschiedenen Codierungsschemata gespeichert wird. Als Beispiele dienen wiederum die Zeichen A, € und ℵ.

BINÄRZAHL	WIRD GESPEICHERT IN DER FORM ...		
	UTF-8	UTF-16LE	UTF-16BE
0100 0001	0100 0001 (1 x 1 Byte)	0000 0000 0100 0001 (1 x 2 Byte)	0100 0001 0000 0000 (1 x 2 Byte)
0010 0000 1010 1100	1110 0010 1000 0010 1010 1100 (3 x 1 Byte)	0010 0000 1010 1100 (1 x 2 Byte)	1010 1100 0010 0000 (1 x 2 Byte)
0101 1101 0000	1101 0111 1001 0000 (2 x 1 Byte)	0000 0101 1101 0000 (1 x 2 Byte)	1101 0000 0000 0101 (1 x 2 Byte)

Die Codierungsform

Erst durch die Codierungsform lassen sich die Zeichen als Daten im Computer darstellen. In der Codierungsform sind die Codenummern eines Zeichensatzes jeweils einer oder mehreren Code-Einheiten zugeordnet. Eine *Code-Einheit* ist ein Speicherplatz innerhalb der Computer-Architektur mit einer bestimmten binären Länge (z. B. 1 Byte, also 8 Bit). Eine *Code-Einheit-Sequenz* ist der Platz, der für die Repräsentation und Speicherung einer Codenummer verwendet wird. Die Anzahl an Code-Einheiten pro Zeichen muss nicht unbedingt gleich sein, vielmehr gibt es Codierungsformen *fester* und *variabler Länge.* Für den Unicode-Zeichensatz etwa gibt es mehrere Codierungsformen: ein Beispiel für eine Codierungsform fester Länge ist *UCS-2,* Beispiele für Codierungsformen variabler Länge sind *UTF-8* (Sequenzen von ein bis vier mal 1 Byte) und *UTF-16* (Sequenzen von ein bis zwei mal 2 Byte). Die Standardcodierungsformen für *Unicode 3.0* und XML sind *UTF-16* und *UTF-8.*

Das Codierungsschema

In einem Codierungsschema wird einer Code-Einheit eine bestimmte Bytereihenfolge zugeordnet. Bei Code-Einheiten mit einer Länge von 1 Byte wird das eine Byte einfach auf ein Byte mit demselben Wert abgebildet. Bei längeren Code-Einheiten jedoch ist je nach Plattform möglicherweise eine Umkehrung der Bytereihenfolge erforderlich. Für die Codierungsform *UTF-16* etwa gibt es aus diesem Grund die beiden Codierungsschemata *UTF-16LE* und *UTF-16BE,* wobei *LE* für *Little Endian byte order* steht *(Byteordnung: niederwertiges Byte zuerst,* auch *Intel-Format)* und *BE* für *Big Endian byte order (Byteordnung: höherwertiges Byte zuerst,* auch *Motorola-Format).* Das Codierungsschema ist Hardware- und somit Plattform-abhängig. Auf Unix- und Windows-Systemen ist der Standard *UTF-16LE.*

UTF-8 und UTF-16

UTF-8 und *UTF-16* sind zwei *Codierungsformen* des Unicode-Zeichensatzes und der Standard für XML-Dokumente. Beide verwenden Code-Einheit-Sequenzen variabler Länge, d. h., sie belegen für verschiedene Zeichen unterschiedlich viel Platz.

Bei *UTF-8* sind das ein bis vier mal 1 Byte: Die ersten 128 Byte (0 bis 127) sind in jeweils einem Byte codiert und somit identisch mit dem ASCII-Zeichensatz, weitere nicht ideographische Zeichen belegen die Plätze 128 bis 2047 mit 2 x 1 Byte und Ideogramme (in der Hauptsache chinesische, japanische und koreanische Schriftzeichen) belegen die momentan hintersten Plätze mit 3 x 1 Byte. Wenn in der Zukunft neue Zeichen über der Codenummer 65 535 in den Unicode-Zeichensatz Eingang finden, werden diese mit 4 x 1 Byte codiert. Wegen der ASCII-Konformität kann eine UTF-8-Datei, die nur englischen Text enthält, auch mit einem ASCII-Editor bearbeitet werden oder, anders ausgedrückt, eine ASCII-Datei kann in der Text- oder XML-Deklaration als UTF-8-Datei ausgegeben werden (allerdings nur „echter“ (US)ASCII-Text mit dem Basiszeichenvorrat des englischen Alphabets).

Bei *UTF-16* hingegen ist jedes Zeichen auf mindestens 2 Byte abgebildet, genauer gesagt auf Sequenzen von ein bis zwei mal 2 Byte. Der Bereich von 2 x 2 Byte ist momentan noch frei (um den Unicode-Zeichensatz auch für eine ferne Zukunft mit vielen, vielen neuen Schriftzeichen noch universell zu halten), somit ist die variable Länge hier rein theoretischer Natur. Eine in *UTF-16* gespeicherte Datei ist, was die Codierungsform anbelangt, nicht ASCII-konform und somit von einem ASCII-Editor nicht lesbar (obgleich die Codenummern dieselben sind wie beim ASCII-Zeichensatz).

Wichtig: Aus dem Aufbau der beiden Codierungsformen ergibt sich, dass *UTF-8* eher für nicht ideographische Sprachen geeignet ist, da eine solche in *UTF-8* gespeicherte Datei um einiges kleiner ist. Entsprechend eignet sich *UTF-16* eher für Sprachen, die sich aus Ideogrammen zusammensetzen.

Bei *UTF-16* ist zudem das *Codierungsschema* zu beachten, welches die Byteordnung *LE (Little Endian, niederwertiges Byte zuerst)* bzw. *BE (Big Endian, höherwertiges Byte zuerst)* berücksichtigt. Die Byteordnung ist Hardware-abhängig. *LE* oder *BE* entscheidet darüber, welches Byte als erstes (d. h. im unteren Speicherplatz) gespeichert wird (siehe auch den Abschnitt *Das Codierungsschema*). Ein XML-Parser kann selbst erkennen, ob eine ihm vorgelegte Datei in *UTF-8-*, *UTF-16LE-* oder *UTF-16BE*-codiert ist. Denn eine UTF-16-Datei beginnt mit einem speziellen Zeichen, das als Kennzeichen für die Byteordnung dient: Bei *UTF-16LE* ist das das Zeichen mit der Hexadezimalnummer 0xFF FE, bei *UTF-16BE* ist es 0xFE FF.

Die Codebereiche

Der gesamte Unicode-Zeichensatz erstreckt sich im Bereich 0x0 bis 0x10 FF FF (Hexadezimalschreibweise) bzw. 0 bis 1 114 111 (Dezimalschreibweise). Er ist in 17 Ebenen eingeteilt mit jeweils 65 536 Codewerten. Die erste davon ist die so genannte *BMP (Basic Multilingual Plane, Mehrsprachenbasisebene)* und deckt alle modernen Sprachen der Welt ab. Diese Ebene ist wiederum in mehrere Teilmengen untergliedert, die jeweils einen bestimmten Codebereich umfassen. Die nachfolgende Tabelle bietet einen Überblick über den Inhalt dieser Codebereiche.

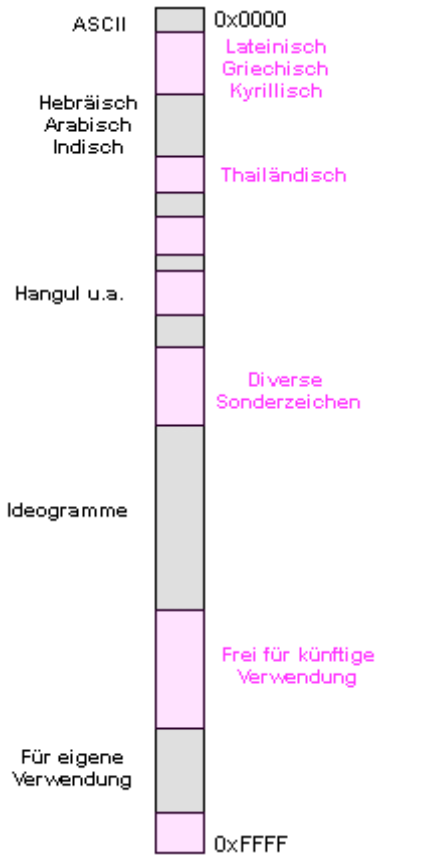

Abbildung 7.6: Übersicht über die Codebereiche der ersten Ebene.

CODEBEREICH		INHALT	NATION
Hex.	Dez.		
0000–007F	00000–00127	C0-Steuerzeichen und die grundlegenden Zeichen der lateinischen Schrift	Diverse
0080–00FF	00128–00255	C1-Steuerzeichen und Ergänzungszeichen zur lateinischen Schrift	Diverse
0100–017F	00256–00383	Lateinische Schriftzeichen, erweiterte Version A	Diverse
0180–024F	00384–00591	Lateinische Schriftzeichen, erweiterte Version B	Diverse
0250–02AF	00592–00687	Internationales phonetisches Alphabet	Diverse
02B0–02FF	00688–00767	Weitere Artikulationszeichen	Diverse
0300–036F	00768–00879	Unterscheidungszeichen für die Kombination mit anderen Zeichen	Diverse
0370–03FF	00880–01023	Griechisch und Koptisch	Griechenland
0400–04FF	01024–01279	Kyrillisch	Russland, Ukraine u. a.
0530–058F	01328–01423	Armenisch	Armenien
0590–05FF	01424–01535	Hebräisch	Israel
0600–06FF	01536–01791	Arabisch	Diverse
0900–097F	02304–02431	Devanagari	Indien
0980–09FF	02432–02559	Bengali	Indien
0A00–0A7F	02560–02687	Gurmukhi (Punjabi)	Indien u. a.
0A80–0AFF	02688–02815	Gujarati	Indien
0B00–0B7F	02816–02943	Oriya	Indien
0B80–0BFF	02944–03071	Tamil	SriLanka, Indien
0C00–0C7F	03072–03199	Telegu	Indien
0C80–0CFF	03200–03327	Kannada (Kanarese)	Indien

CODEBEREICH		INHALT	NATION
Hex.	Dez.		
0D00–0D7F	03328–03455	Malayalam	Indien
0D80–0DFF	03456–03583	Sinhalese	Sri Lanka
0E00–0E7F	03584–03711	Thai	Thailand
0E80–0EFF	03712–03839	Laotisch	Laos
0F00–0FBF	03840–04031	Tibetisch	Tibet
10A0–10FF	04256–04351	Georgisch	Georgien
1100–11FF	04352–04607	Hangul Jamo	Korea
1200–137C	04608–04988	Äthiopisch	Äthiopien
13A0–13FF	05024–05119	Tsalagi (Cherokee)	U.S.A.
1401–1676	05121–05750	Silben der Sprachen der kanadischen Ureinwohner	Cree u. a.
1680–169C	05760–05788	Ogham (Altes Irish)	Irland
1700–1759	05888–05977	Myanmar (Birmanisch)	Myanmar
1E00–1EFF	07680–07935	Lateinische Schriftzeichen, zusätzliche erweiterte Version	Diverse
1F00–1FFF	07936–08191	Griechische Schriftzeichen, erweiterte Version	Griechenland
2000–206F	08192–08303	Allgemeine Interpunktionszeichen	
2070–209F	08304–08351	Hoch und tief gestellte Zeichen	
20A0–20CF	08352–08399	Währungszeichen	
20D0–20FF	08400–08447	Unterscheidungszeichen für die Kombination mit Symbolen	
2100–214F	08448–08527	Buchstabenähnliche Symbole	
2150–218F	08528–08591	Nummerierungsziffern	
2190–21FF	08592–08703	Pfeile	
2200–22FF	08704–08959	Mathematische Operatoren	

CODEBEREICH		INHALT	NATION
Hex.	Dez.		
2300–23FF	08960–09215	Diverse technische Zeichen	
2400–243F	09216–09279	Bilder für Steuerzeichen	
2440–245F	09280–09311	OCR (Optical Character Recognition, optische Zeichenerkennung)	
2460–24FF	09312–09471	Ziffern und lateinische Buchstaben in Klammern und Kreisen	
2500–257F	09472–09599	Symbole für Gitterzeichnungen	
2580–259F	09600–09631	Flächenelemente	
25A0–25FF	09632–09727	Geometrische Formen	
2600–26FF	09728–09983	Verschiedene Symbole	
2700–27BF	09984–10175	Dingbats (Schmuckzeichen und Clipart)	
3000–303F	12288–12351	CJK-Symbole und Interpunktion	CJK *
3040–309F	12352–12447	Hiragana	Japan
30A0–30FF	12448–12543	Katakana	Japan
3100–312F	12544–12591	Bopomofo (Phonetisch)	Taiwan
3130–318F	12592–12687	Hangul-Kompatibilität/Jamo	Korea
3200–32FF	12800–13055	CJK-Buchstaben und -Monate in Klammern und Kreisen	CJK
3300–33FF	13056–13311	CJK-Kompatibilität	CJK
4E00–9FFF	19968–40959	CJK-Ideogramme	CJK
AC00–D7A3	44032–55203	Hangul-Silben	Korea
D800–DB7F	55296–56191	High Surrogates	
DB80–DBFF	56192–56319	Surrogate für eigene Verwendung	
DC00–DFFF	56320–57343	Weitere Surrogate	
E000–F8FF	57344–63743	Bereich für eigene Verwendung	
F900–FAFF	63744–64255	CJK-Kompatibilitäts-Ideogramme	CJK
FB00–FB4F	64256–64335	Alphabetische Präsentationsformen (Ligaturen und andere)	
FB50–FDFF	64336–65023	Arabische Präsentationsformen, Variante A	
FE20–FE2F	65056–65071	Halbe Zeichen für die Kombination mit anderen	
FE30–FE4F	65072–65103	CJK-Kompatibilitätsformen	CJK
FE50–FE6F	65104–65135	Varianten für kleine Formen	
FE70–FEFF	65136–65279	Arabische Präsentationsformen, Variante B	
FF00–FFEF	65280–65519	Formen halber und voller Breite	
FFF0–FFFF	65520–65535	Besondere Zeichen	

* CJK steht für Chinesisch/Japanisch/Koreanisch bzw. China/Japan/Korea

Die Sprachparameter für und in XML

An dieser Stelle sollen alle für XML relevanten Spracheinstellungen noch einmal zusammengefasst werden, die teilweise bereits an anderen Stellen des Buchs ausführlich erläutert worden sind.

Ländereinstellung und Tastaturtreiber

Welche Zeichen und Zeichensätze auf Ihrem System verfügbar sind, hängt zunächst von Ihren Systemeinstellungen ab (unter *Windows* mit **SYSTEMSTEUERUNG/LÄNDEREINSTELLUNGEN** einstellbar), denn diese bestimmen, welche Codepages bzw. Unicode-Teilmengen auf Ihrem Computer installiert werden. Selbst wenn Sie Ihre XML-Dokumente in Unicode speichern oder korrekte Zeichenreferenzen verwenden, heißt das noch lange nicht, dass die entsprechenden Zeichen auch angezeigt werden. Denn es sind immer nur die Schriftzeichen lesbar, die Ihre Spracheinstellungen bzw. die Einstellungen des Client-Systems unterstützen. Für den Webdesigner bedeutet dies, dass Sie immer auch wissen sollten, für welchen Zielsprachraum Sie eine XML-Seite erstellen.

Welche Zeichen Sie über Ihre Tastatur direkt eingeben können, hängt von einem speziellen Programm, dem so genannten *Tastaturtreiber* ab (unter *Windows* mit **SYSTEMSTEUERUNG/TASTATUR** einstellbar). Bei der Eingabe von Text sendet jede Taste eine bestimmte Codenummer an den Computer. Die Zuordnung der Codenummern zu den Tasten erfolgt durch diesen Tastaturtreiber.

Die Texteditoren

Viele Texteditoren machen es von diesen Einstellungen abhängig, welche Dateiformate sie anbieten. Mit der Ländereinstellung für die deutsche Sprache wird in der Regel der Zeichensatz *ISO-8859-1* oder ein verwandter Zeichensatz verwendet (auf Windows-Systemen etwa *Cp1252,* auf Macintosh-Systemen *MacRoman,* die jedoch beide von regulären XML-Parsern nicht unterstützt werden und somit für XML-Dokumente nicht zu empfehlen sind).

Die Schriften

Welche Zeichen auf Ihrem Computer angezeigt werden, hängt letztlich von den verfügbaren Schriften (Fonts) ab. Standardmäßig werden nur Schriften installiert, die für die von Ihnen gewählten Ländereinstellungen und Tastaturtreiber erforderlich sind. Bei Bedarf können Sie jedoch selbst weitere Schriften installieren.

Der XML-Zeichensatz

Der XML-Zeichensatz ist *Unicode* und somit muss ein XML-Parser jedes Unicode-Zeichen unterstützen. Wenn Sie ein XML-Dokument verwenden, das in einem anderen Zeichensatz gespeichert ist, müssen Sie dies in der XML-Deklaration angeben.

Der aktuelle Zeichensatz

Der aktuelle Zeichensatz ist der Zeichensatz, den Sie in der XML-Deklaration des XML-Dokuments angegeben haben. In diesem Zeichensatz wurde das Dokument codiert und gespeichert. Wenn Sie Zeichenreferenzen verwenden, beziehen sich diese nicht etwa auf die Codenummern des aktuellen Zeichensatzes, sondern immer auf die des Unicode-Zeichensatzes.

Lesen Sie nach ...

- Wie Sie einige Unicode-Schriften installieren, erfahren Sie im Abschnitt *Unicode-Schriften.*
- Wollen Sie das Thema Unicode-Zeichensatz vertiefen? Dann lesen Sie oben den Abschnitt *Unicode.*
- Zeichensätze im Allgemeinen und die ISO-Zeichensätze im Besonderen werden im Abschnitt *Zeichensätze in XML* erläutert.

Lesen Sie nach ...

- Wie Sie den aktuellen Zeichensatz deklarieren, erfahren Sie in Kapitel 6, Abschnitt *Die XML-Deklaration.*
- Die Verwendung von Zeichenreferenzen wird in Kapitel 6, Abschnitt *Zeichenreferenzen* beschrieben.

Lesen Sie weiter ...

- Wie Sie den Zeichensatz einer externen Entity deklarieren, wird in Kapitel 8, Abschnitt *Entity-Konzepte, Die Textdeklaration* erläutert.

Die XML-Deklaration

Ist Ihr XML-Dokument nicht in *Unicode* codiert und gespeichert (siehe auch den Abschnitt *UTF-8 und UTF-16*), sondern in einem anderen Zeichensatz (etwa *ISO-8859-1*), so teilen Sie dies dem XML-Parser in der XML-Deklaration mit.

Die Textdeklaration

Wenn Sie in Ihrem XML-Dokument auf eine externe Entity verweisen, die nicht in Unicode codiert ist, müssen Sie am Anfang der Entity eine Textdeklaration einfügen, die den Entity-Zeichensatz benennt.

Zeichenreferenzen

Zeichenreferenzen geben Ihnen die Möglichkeit, im XML-Dokument über die Codenummer im Unicode-Zeichensatz auf jedes existierende Schriftzeichen zu verweisen

Das Attribut xml:lang

Mithilfe des Attributs `xml:lang` können Sie den Inhalt eines Elements einer bestimmten Sprache zuordnen. Zwar hat das von der XML-Spezifikation vorgegebene Attribut zunächst gar keine Auswirkung auf das XML-Dokument, doch lässt es sich in XML-verarbeitender Software oder in einem XSLT-Stylesheet etwa dazu nutzen, die fremdsprachliche Information herauszufiltern, korrekt zu indizieren oder einer geeigneten Rechtschreibprüfung zu unterziehen.

Angenommen, Sie möchten ein mehrsprachiges Wörterbuch zu einem bestimmten Thema erstellen. Dafür eignet sich XML ausgezeichnet. Dem Originalbegriff sowie den verschiedenen Übersetzungen ordnen Sie anhand des Attributs `xml:lang` ihre Sprache zu, so dass Sie später das jeweils benötigte Sprachenpaar herausfiltern können.

Für die Verwendung der Attributwerte – also der Sprachencodes – gibt es verschiedene Prioritäten:

1. Zunächst suchen Sie die gewünschte Sprache im ISO-639-Standard *(Codes für die Repräsentation von Sprachenbezeichnungen)* von 1988 für 2-stellige Sprachencodes. Dieses ist die wichtigste Quelle und in den meisten Fällen ausreichend.

   Die 2-stelligen Codes reichen nicht für alle Sprachen aus, so dass der ISO-639-Standard inzwischen aktualisiert worden ist, in ISO 639-1 für 2-stellige Codes und ISO 639-2 für 3-stellige. Die Verwendung der 3-stelligen Codes für XML ist noch umstritten, wird aber wohl früher oder später akzeptiert werden.
2. Deckt der ISO-639-Standard eine Sprache oder einen Dialekt nicht ab, dürfen Sie auch die von IANA registrierten Sprachencodes verwenden, die mit *i-* oder *I-* beginnen.
3. Sie haben sogar die Möglichkeit, selbst definierte oder firmenspezifische Sprachencodes zu verwenden, vorausgesetzt Sie stellen das Präfix *x-* oder *X-* voran.

Wenn es von einer Sprache verschiedene Varianten gibt (wie etwa britisches oder amerikanisches Englisch), können Sie zusätzlich noch einen Nebencode spezifizieren, etwa `xml:lang="en-EN"` oder `xml:lang="en-US"`. Der Nebencode wird in Großbuchstaben geschrieben und richtet sich nach dem ISO-3166-Standard, *Codes für die Repräsentation von Länderbezeichnungen.*

✔ Tipps

- Die Codes für die Repräsentation von Sprachenbezeichnungen finden Sie auf der Website *http://lcweb.loc.gov/standards/iso639-2/langhome.html* und der FTP-Site *ftp://ftp.std.com/obi/Standards/ISO/ISO_639*.
- Weitere Sprachencodes sind bei IANA registriert (sogar Klingonisch!). Sie finden sie unter *ftp://ftp.isi.edu/in-notes/iana/assignments/languages/* bzw. *http://www.isi.edu/in-notes/iana/assignments/languages/*.
- Die Nebencodes für Sprachvarianten finden Sie unter der URL-Adresse *ftp://ftp.std.com/obi/Standards/ISO/ISO_3166* aufgelistet.

code.xml

```
<fachbegriff thema="xml">
   <original xml:lang="en">Extensible
   Stylesheet Language</übersetzung>
   <übersetzung xml:lang="de">Erweiter-
   bare Auszeichnungssprache
   </übersetzung>
   <übersetzung xml:lang="fr">Langage de
   balisage extensible</übersetzung>
   <übersetzung xml:lang="ru">
   </übersetzung>
</fachbegriff>
```

Abbildung 7.7: Die Syntax des Attributs `xml:lang` ist genauso wie bei anderen Attributen.

Lesen Sie weiter ...

- Die Deklaration von Attributen in der DTD ist in Kapitel 8, Abschnitt *Attributlistendeklarationen* erläutert.
- Wie Sie den Attributtyp CDATA zuweisen, erfahren Sie ebenfalls in Kapitel 8, Abschnitt *Attributtyp CDATA*.

Lesen Sie nach ...

- Die meisten 2-stelligen Sprachencodes nach ISO 639 finden Sie auch im vorliegenden Kapitel im Abschnitt *Welcher Zeichensatz für welche Sprache?*