

Die erweiterbare Markierungssprache XML

1. Einleitung

Eine Markierungssprache (markup language) dient dazu, Textdateien mit zusätzlichen Informationen anzureichern. Die verbreitete Markierungssprache HTML (hypertext markup language) erlaubt es z.B., bestimmte Teile einer Textdatei als Überschriften und andere Teile als normale Absätze zu kennzeichnen, z.B. so:

```
1 <H1>1. Über den Umgang mit Drachen</H1>
2
3 <P ALIGN="left">
4   Allgemein unterscheidet man zwischen Drachen <B>erster</B> Ordnung
5   und Drachen <B>zweiter</B> Ordnung.
6 </P>
7
8 <H2>1.1. Drachen erster Ordnung</H2>
9 <P ALIGN="left">
10  Drachen <B>erster</B> Ordnung gibt es nicht. Deshalb sind sie in
11  der Natur sehr schwer zu finden und werden von vielen Liebhabern sehr
12  geschätzt.
13 </P>
14
15 <H2>1.2. Drachen zweiter Ordnung</H2>
16 <P ALIGN="left">
17  Drachen <B>zweiter</B> Ordnung gibt es auch nicht, aber aus ganz
18  anderen Gründen als Drachen erster Ordnung.
19 </P>
```

Die Markierungen (tags) in den spitzen Klammern drücken unter anderem aus, dass der Text "1. Über den Umgang mit Drachen" in Zeile 1 ein Überschrift der Stufe 1 ist (H1 wie header 1) und dass die Zeilen 4 und 5 einen Abschnitt (P wie paragraph) bilden, der linksbündig ausgerichtet werden soll (und nicht rechtsbündig oder zentriert).

Die Markierungssprache HTML hat (neben dem Vorteil ihrer weiten Verbreitung) folgenden Nachteil: Sie erledigt zwei Aufgaben auf einmal, die man häufig besser getrennt erledigen sollte. Mit HTML-Markierungen beschreibt man zum einen, aus welchen Teilen ein Text besteht (z.B. aus Überschriften verschiedener Stufe, normalen Abschnitten etc.). Gleichzeitig legt man damit fest, wie diese Teile z.B. auf einem Bildschirm dargestellt werden sollen, z.B. linksbündig oder fett etc.

HTML-Markierungen haben grosse Ähnlichkeit mit Klammern. Z.B. entspricht die Markierung <H1> einer öffnenden Klammer und </H1> ist die zugehörige schliessende Klammer. Leider ist es nach den Regeln der Sprache HTML erlaubt, bestimmte schliessende Markierungen wegzulassen. Eine syntaktische Prüfung von HTML-Dateien wird dadurch so stark erschwert, dass man in der Praxis häufig auf sie verzichtet. Ergebnis: Viele HTML-Dateien enthalten syntaktische Fehler, die zu falschen Darstellungen führen, aber nicht automatisch lokalisiert werden.

2. Die erweiterbare Markierungssprache XML

Die Sprache XML (extensible markup language) ist deutlich mächtiger als HTML. Sie erlaubt es, die abstrakte Struktur von Texten, die konkreten Texte selbst und die Regeln für ihre Darstellung z.B. auf einem Bildschirm getrennt voneinander zu erstellen und zu verwalten. Davon versprechen sich viele Anwender erhebliche Vorteile.

Um einen mit XML strukturierten Text zu erstellen, kann man z.B. so vorgehen:

1. Man erstellt eine Datentyp-Definition (eine .dtd-Datei). Darin legt man z.B. fest, dass die später


```

45     <DIV STYLE="color:red; font-size:18pt; font-family:Times">
46         <xsl:value-of select="Adressen-Liste/Name-der-Liste" />
47     </DIV>
48
49     <!-- Die einzelnen Unterlisten ausgeben: -->
50     <xsl:for-each select="Adressen-Liste/Unterliste">
51
52         <!-- Zuerst die Ueberschrift ausgeben: -->
53         <DIV STYLE="color:blue; font-size:14pt; font-family:Times">
54             <xsl:value-of select="Name-der-Unterliste" />
55         </DIV>
56
57         <!-- dann die einzelnen Adressen einer Unterliste ausgeben: -->
58         <xsl:for-each select="Adresse">
59
60             <!-- Zuerst den Vornamen und den Namen ausgeben: -->
61             <DIV STYLE="color:green; font-size:12pt; font-family:Times">
62                 <SPAN STYLE="width=100">
63                     <xsl:value-of select="Vorname" />
64                 </SPAN>
65                 <SPAN STYLE="width=100">
66                     <xsl:value-of select="Nachname" />
67                 </SPAN>
68
69                 <!-- dann die einzelnen Telefon-Nrn einer Adresse ausg.: -->
70                 <xsl:for-each select="Fon">
71                     <SPAN STYLE="width=110">
72                         <xsl:value-of />
73                     </SPAN>
74                 </xsl:for-each>
75             </DIV>
76         </xsl:for-each>
77     </xsl:for-each>
78
79 </xsl:template>
80 </xsl:stylesheet>

```

Die **fetten Hervorhebungen** sind in der Datei `AdrList_A.xsl` nicht vorhanden. Sie wurden hier angebracht, um die Verbindung zur Datentyp-Definition `AdrList.dtd` deutlich zu machen.

Das "A" im Datei-Namen `AdrList_A.xsl` soll andeuten, dass auch noch ganz andere Ausgabeformate für Adressen-Liste(n) denkbar sind, die man z.B. in Dateien namens `AdrList_B.xsl`, `AdrList_C.xsl` etc. beschreiben könnte.

`<DIV>` und `` sind HTML-Markierungen, die man auch in einer XSL-Stildefinition verwenden kann. Mit diesen Markierungen kann man den Schrifttyp, die Grösse, die Farbe etc. einer Ausgabe festlegen. Die Markierung `<DIV>` bewirkt vor der Ausgabe einen Zeilenwechsel, die Markierung `` nicht.

Statt durch eine `.xsl`-Datei kann man die Ausgabe auch durch eine `.css`-Datei (cascading style sheet) beschreiben. Hier eine `.css`-Datei, die zur Datendefinitionsdatei `AdrList.dtd` passt:

```

81 <!-- - - - - -
82 Datei AdrList_A.css (fuer den Datentyp Adressen-Liste, der in der Datei
83 AdrList.dtd definiert wird.
84 - - - - -
85
86 Adressen-Liste {
87     background-color: lime;
88 }
89
90 Vorname {
91     color: red;
92     background-color: yellow;

```



```

145     <Fon>          0172-3456-7890 </Fon>
146   </Adresse>
147
148   <Adresse>
149     <Nachname> Seidelsdorf </Nachname>
150     <Vorname>  Wilhelmine </Vorname>
151     <Fon>      030-883-7689 </Fon>
152     <Fon>      040-674-9215 </Fon>
153   </Adresse>
154
155   <Adresse>
156     <Nachname> Trossman </Nachname>
157     <Vorname>  Fridolin </Vorname>
158     <Fon>      030-765-1243 </Fon>
159     <Fon>      040-394-5643 </Fon>
160   </Adresse>
161
162   </Unterliste>
163 </Adressen-Liste>

```

Wenn man sich diese Datei `AdrList01.xml` mit dem IE anzeigen lässt, erscheint ungefähr folgendes auf dem Bildschirm (für den Fall, dass Sie eine schwarz-weiß-Kopie dieses Papiers lesen, steht rechts neben jeder Zeile, in welcher Farbe sie auf dem Bildschirm erscheint: schwarz, rot, blau oder grün):

Dieser Text wird zuerst ausgegeben!

Adressliste von Ottokar Meyerbeer

Persoenliche Bekannte und Freunde

Heinz-Dieter Schmidt 030-4504-2265

Gertrude Mueller 030-883-7689 040-674-9215 0172-3456-7890

Berufliche Kontakte

Kunigunde Aschenhus 030-4504-2265 0172-3456-7890

Wilhelmine Seidelsdorf 030-883-7689 040-674-9215

Fridolin Trossman 030-765-1243 040-394-5643

schwarz

rot

blau

grün

grün

blau

grün

grün

grün

4. Programme zum Bearbeiten von XML-Dateien

Im Prinzip kann man `.dtd`-Dateien, `.xsl`-Dateien und `.xml`-Dateien mit jedem gewöhnlichen ASCII-Editor erstellen und verändern. Allerdings ist es ziemlich mühsam, die vielen Markierungen alle von Hand zu tippen.

Es gibt heute (Ende 2010) zahlreiche XML-Editoren. Alle erleichtern einem (mehr oder weniger) das Erstellen von XML-Dateien. Einige unterstützen auch das Erstellen und Verändern von DTD-, von XSL-, von XSD- (XML-Schema Dateien) und weiteren Dateien.

Umsonst zu haben sind unter andern der XML-Editoren Serna (siehe www.syntext.com) und der XML Notepad 2007 von Microsoft. Weitere interessante Internet-Adressen und Hinweise findet man unter der Adresse www.xmlguru.de.