

Einführung in XML

von Julian Bart

Über diese Einführung

Diese Einführung soll Ihnen einen groben, ersten Überblick über XML, dem designierten Nachfolger der Seitenbeschreibungssprache HTML 4.0, vermitteln und Ihnen einen ersten Einstieg ermöglichen. Sie erhebt keinen Anspruch auf Vollständigkeit.

XML verfügt über verbesserte Möglichkeiten der Portierung und der Erweiterung. Da eine weitreichende Vernetzung und der mobile Datenaustausch in immer mehr Geräten Einzug halten, seien es nun Spielekonsolen oder Handheld-Computer, steigen damit die Anforderungen an die Sprache, mit der diese Ziele erreicht werden sollen. Das datenbankfreundliche und einfach portierbare XML (auch XHTML, "Extended Hypertext Markup Language") soll da in die Bresche springen.

Inhalt

- Einführung
- Kapitel Eins: Aufbau einer XML-Datei
- Kapitel Zwei: Schreiben einer XML-Datei
- Kapitel Drei: Darstellung einer XML-Datei
- Kapitel Vier: Aufbau einer XSL-Datei
- Kapitel Fünf: Schreiben einer XSL-Datei
- Kapitel Sechs: Verwendung von Links

Der Autor

Der Autor dieser Einführung, Julian Bart, ist Informatikstudent und wissenschaftliche Hilfskraft im Rechenzentrum der Universität Stuttgart. Er ist Betreiber der Internetseite "SelfXML" (<http://www.selfxml.de/>), auf der auch ein Forum zu diesem Thema zu finden ist.

Sie erreichen ihn unter der Mailadresse author@selfxml.de

Einführung in XML

von Julian Bart

Einführung

XML tritt meistens in 2 Dateien auf:

- Die XML-Datei enthält die Textinformationen, die von Tags eingeschlossen sind, deren Namen Sie selber vergeben können.
- Die XSL-Datei enthält die Gestaltungsmerkmale für die Textinformationen der XML-Datei.

Der Vorteil von XML ist, daß Sie ein XSL-File für mehrere XML-Dateien verwenden können. Das ist vor allem von Vorteil, wenn Sie viele Seiten machen wollen, deren Aussehen aber einheitlich bestimmt werden soll. Wenn man bei HTML die Farbe des Textes auf 20 Seiten ändern will, muß man dies auf jeder einzeln machen. Bei XML muß man bloß das eine XSL-File ändern, um die Änderung auf allen Seiten zu bewirken. Ein einheitliches Bild ist gerade bei größeren Produktlandschaften interessant.

XML & Datenbanken

Auch dieser Bereich ist sehr interessant, nur im Moment noch nicht ganz fertig entwickelt. Sowohl Microsoft als auch IBM arbeiten an einer Lösung zum Export von Daten aus Datenbanken direkt in eine XML-Datei. Hier ist es von großem Vorteil, daß man den Tags einen eigenen Namen geben kann, wie z.B. <Name> und nicht auf die bei HTML vorgegebenen Tags wie beispielsweise <H1> angewiesen ist.

XML - Data Isles

XML-Daten können auch direkt in HTML-Dateien untergebracht werden. Auf diese wird dann direkt zugegriffen. Im Moment haben wir nicht vor, näher auf diese Art der XML-Verwendung einzugehen. Vermutlich können wir sie früher oder später nicht mehr ignorieren.

Einführung in XML

von Julian Bart

Kapitel Eins: Aufbau einer XML-Datei

Ein XML-File beginnt mit der Definition, welche XML-Version benutzt wird (bei MSIE 5 nicht nötig). Diese ist zur Zeit noch die Version 1.0, Auf Weiterentwicklung bleibt abzuwarten.

```
<?xml version="1.0" ?>
```

Neben dem verwendeten Zeichensatz kann in dieser Definition auch angegeben werden, ob eine Document Type Definition (kurz "DTD") existiert oder nicht. Dafür wird der Begriff "**standalone**" benutzt. Diese Definition wird beim IE 5 ebenfalls nicht benötigt.

Der Vorteil der Benutzung von DTD-Dateien liegt vor allem in der Vorgabe von Standard-Attributen, die dann automatisch eingesetzt werden.

```
<?xml version="1.0" standalone="yes" ?>
```

In der nächsten Zeile gibt es eine weitere, sehr wichtige Einstellung zu erledigen, wenn man eine XSL-Datei verwendet: Man muß noch die Datei angeben, in der sich der Style Sheet befindet. Diese sollte die Endung XSL haben, was aber nicht zwingend notwendig ist. Diese beiden Tags sind selbstständig, sie müssen also nicht mehr geschlossen werden.

```
<?xml version="1.0" standalone="yes" ?>
```

```
<?xml-stylesheet type="text/xsl" href="Datei.xsl" ?>
```

Natürlich gibt es noch viele andere Einstellungsmöglichkeiten im Dokumentkopf, auf die wir hier aber nicht näher eingehen wollen. Die grundlegenden Möglichkeiten haben Sie nun kennengelernt.

Bevor man beginnt, muß noch ein Root-Element ausgewählt werden, also das Element, das alle anderen Elemente beinhaltet. Hiervon darf es nur ein Einziges geben, dessen Namen aber auch frei gewählt werden darf.

Einführung in XML

von Julian Bart

Kapitel Zwei: Schreiben einer XML-Datei

Eine XML-Datei besteht aus verschiedenen Elementen, deren Namen Sie von Ihnen gewählt werden können und deren Inhalt aus Text besteht. Ein Element kann auch Attribute haben. Aus mehreren Elementen ergibt sich dann eine Struktur, die sich der Autor bereits vorher überlegen sollte. Diese Struktur beginnt mit dem Root-Element, das Ausgangspunkt für alle anderen Elemente ist und sie beinhaltet.

Eine wesentliche Erleichterung beim Schreiben ist das kostenlos von Microsoft zur Verfügung gestellte XML-Notepad. Um dieses zu benutzen sollte jedoch der IE 5 installiert sein, da sonst das Notepad alle Elementnamen in Grossbuchstaben ändert und dies im XSL-File berücksichtigt werden muß. Natürlich gibt es noch andere Hilfsprogramme, wobei fast jedes so seine Feinheiten hat, die verhindern, das das Ergebnis im IE 5 dargestellt werden kann.

Beim Benennen der Elementnamen muß darauf geachtet werden, daß XML die Groß- und Kleinschreibung unterscheidet. Also ein mit <NAME> geöffnetes Element, wird nicht durch das Tag </Name> wieder geschlossen. Wichtig ist auch das jedes geöffnete Element unbedingt wieder geschlossen wird, da sonst sofort eine Fehlermeldung erscheint. Es muß auch darauf geachtet werden, daß die Struktur genau eingehalten wird und die geöffneten Tags an den richtigen Stellen wieder geschlossen werden.

Ein kleines Beispiel:

```
<?xml-version="1.0"?>
<?xml-stylesheet type="text/xsl" href="Autoren.xsl"?>
<Personen>
  <Autor ID="1"> <!-- Hier steht ein Kommentar. -->
    <Vorname>Frank</Vorname>
    <Nachname>Mueller</Nachname>
  </Autor>
  <Autor ID="2">
    <Vorname>Stefan</Vorname>
    <Nachname>Maier</Nachname>
  </Autor>
</Personen>
```

Einführung in XML

von Julian Bart

(...) In diesem Beispiel ist **<Personen>** das Root-Element, das alle Elemente der Datei beinhaltet. Innerhalb des Autoren-Elementes gibt es zwei Elemente mit den Child-Elementen Vorname und Nachname und dem Attribut ID, welches hier zur Durchnummerierung verwendet wird.

Attribute werden im Elementanfangstag angegeben, deren Inhalte werden mit einem Gleichheitszeichen und in Anführungszeichen dargestellt. Wenn man diese XML-Datei im IE 5 öffnet, wird automatisch ihre Struktur in einem Baum dargestellt, da keine XSL-Datei angegeben wurde.

Sicherlich soll öfter mal ein Kommentar eingefügt werden, diesen fügt man zwischen das Start-Tag **<!--** und das End-Tag **-->** (s.obiges Beispiel).

Einführung in XML

von Julian Bart

Kapitel Drei: Darstellung einer XML-Datei

Nachdem man nun seine XML-Datei fertig gestellt hat, will man diese natürlich auch darstellen. Für die erste Ansicht bietet der Internet Explorer 5 eine Standarddarstellung in Form eines Baumes, der einfach die Konstruktion und die Elemente mitsamt Inhalt darstellt, wodurch eine Fehlerprüfung leicht fällt. Auch die Struktur ist sehr gut anzuschauen.

Um die Seite nun entsprechend zu gestalten, kommt XSL zum Einsatz. Natürlich gibt es auch andere Möglichkeiten eine XML-Datei kreativ darzustellen, wie zum Beispiel DSSSL. Diese sind jedoch nicht so interessant, da sie nicht vom Internet Explorer dargestellt werden können.

Auf den folgenden Seiten werden wir auf die Verwendung von XSL eingehen.

Einführung in XML

von Julian Bart

Kapitel Vier: Aufbau einer XSL-Datei

In der XSL-Datei stehen die Anweisungen, wie die Elemente, die in der XML-Datei stehen, dargestellt werden. Es können auch Funktionen wie beispielsweise Schleifen und Sortieren verwendet werden.

Dies ist ein großer Vorteil gegenüber HTML. In der XSL-Datei kann auch noch Text geschrieben werden, der dann einfach ausgegeben wird. Dies ist besonders praktisch bei Überschriften usw. Auch das von HTML bekannte JavaScript ist verwendbar, man muß jedoch beachten, daß es auch ausgeführt wird, jedoch später dazu mehr.

Der Kopf der XSL-Datei hat eine gewisse Ähnlichkeit mit dem einer XML-Datei, da auch er mit der Definition der XML-Version beginnt. Auch hier ist es im Moment noch die Version "1.0". In der nächsten Zeile gibt man dann an, welches Stylesheet man benutzt.

!WICHTIG!

Die Stylesheet-Definition muss am Ende der Datei unbedingt wieder geschlossen werden, da es sonst zu Fehlern kommt.

```
<?xml version="1.0"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
...Stylesheet....
```

```
</xsl.stylesheet>
```

Eine XSL-Datei besteht aus verschiedenen Regeln, welche für jedes Element beschreiben, wie es dargestellt werden soll, wobei Elemente gleichen Namens zusammengefaßt werden.

Den Anfang macht dabei im Idealfall eine Regel, die alle Elemente durchlaufen, da man sie so nicht jedes Mal wiederholen muß. Dieses Element beschreibt man mit "/". Man muß beachten, daß jedes Element erst einmal aufgerufen werden muß, damit es bearbeitet wird. Dafür wird das Tag

```
<xsl:apply-templates/>
```

 benutzt.

Nun wollen wir aber unsere erste Datei erst einmal schreiben.

Einführung in XML

von Julian Bart

Kapitel Fünf: Schreiben einer XSL-Datei

Nun beginnen wir mit dem Erstellen einer XSL-Datei. Als Beispiel soll die Datei des zweiten Kapitels dienen. Hier ist es noch einmal:

```
<?xml-version="1.0"?>

<?xml-stylesheet type="text/xsl" href="Autoren.xsl"?>

<Personen>

  <Autor ID="1"> <!-- Hier steht ein Kommentar. -->

    <Vorname>Frank</Vorname>

    <Nachname>Mueller</Nachname>

  </Autor>

  <Autor ID="2">

    <Vorname>Stefan</Vorname>

    <Nachname>Maier</Nachname>

  </Autor>

</Personen>
```

Wir schauen uns nun zuerst einmal die Struktur der XML-Datei an und stellen fest, daß es im Root-Element "Personen" das Element "Autoren" gibt. Das wiederum enthält zwei Elemente mit dem Namen "Autor". Diese besitzen auch das Attribut "ID". Innerhalb der "Autor"-Elemente gibt es jeweils die Elemente "Vorname" und "Nachname", die Text beinhalten.

Ziel ist es nun, eine Tabelle mit den Überschriften "Vorname" und "Nachname" zu erstellen, in die automatisch die angegebenen Namen geladen werden.

Zuerst der Kopf der XSL-Datei:

```
<?xml version="1.0"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

Dieser Kopf beginnt jede Seite.

Einführung in XML

von Julian Bart

Nun fangen wir mit dem sogenannten Root-Element an. Hier können wir erstmal ein Standardformat angeben. Desweiteren können wir auch schon die Überschriften "Vorname" und "Nachname" angeben, und mit dem Aufbau der Tabelle beginnen.

```
<xsl:template match="/">
```

```
<TABLE border="4pt" >!-- hier wird eine Tabelle mit 4 pt Rahmenstärke geöffnet -->
```

```
<TR STYLE="font-size:16pt; color:red" >!-- Eine Zeile der Tabelle wird geöffnet -->
```

```
<TD>Vorname</TD> <!-- Erster Block mit dem Inhalt Vorname wird gebaut -->
```

```
<TD>Nachname</TD> <!--Zweiter Block mit Nachname -->
```

```
</TR> <!-- Hier wird die erste Zeile der Tabelle beendet. -->
```

```
<xsl:apply-templates/> <!--Hier werden alle Elemente direkt unterhalb aufgerufen, also in diesem Fall "Personen" -->
```

```
</TABLE> <!-- Beenden der Tabelle, die oben aufgerufenen Elemente werden jedoch in die Tabelle eingefügt -->
```

```
</xsl:template>
```

Innerhalb der Tabelle wird das nächste Unterelement, "Personen", aufgerufen, für das wir nun ebenfalls eine Regel schreiben müssen. Da wir uns noch innerhalb der Tabelle befinden, können wir gleich die Tabellenbefehle benutzen.

```
<xsl:template match="Personen" >!-- Hier beginnen wir mit der Personenregel -->
```

```
<xsl:for-each select="Autor" >!--Hier beginnt eine Schleife, alles was hier dazwischen steht, wird für jedes Autor-Element extra ausgeführt -->
```

```
<TR STYLE="font-size:16pt; color:blue" >
```

```
<TD>
```

```
<xsl:value-of select="Vorname"/> <!-- Hier wird der Inhalt von dem Element "Vorname" ingesetzt -->
```

```
</TD>
```

```
<TD>
```

```
<xsl:value-of select="Nachname"/> <!-- und hier von dem zweiten Element -->
```

```
</TD>
```

```
</TR>
```

Einführung in XML

von Julian Bart

```
</xsl:for-each> <!-- Hier endet die Schleife -->
```

```
</xsl:template>
```

Nun ist es fast geschafft: Wir haben eine Schleife erstellt, die für jedes Autor-Element den Vornamen und den Nachnamen ausgibt. Der Vorteil ist hier, daß in der XML-Datei beliebig viele Autor-Elemente verzeichnet sein dürfen, solange sie die Struktur mit Vornamen und Nachnamen einhalten.

Nicht vergessen darf man am Schluss, das Stylesheet Tag wieder zu schliessen.

```
</xsl:stylesheet>
```

Nur mit dem Internet Explorer 5:

Wenn man nun diese Datei als "Autoren.xsl" abspeichert und mit der XML-Datei in ein Verzeichnis stellt, kann man, indem man sie öffnet, das Beispiel anschauen.

Einführung in XML

von Julian Bart

Kapitel Sechs: Verwendung von Links in XML

Ein wichtiger Punkt beim Aufbau einer Seite ist natürlich die Verwendung von Links. Um einen einfachen Link einzubauen, genügt es den von HTML her bekannten Befehl einzubauen. Im Beispiel erstellen wir eine Überschrift mit Link:

```
<xsl:template match="/">
```

```
<H1>Ein Beispiel von <A HREF="http://www.selfxml.de">SelfXML</A> !</H1>
```

```
.....
```

```
</xsl:template>
```

Wir haben nun in das Root-Element einfach eine Überschrift des Formates H1 eingefügt, das den Link zu dieser Seite beinhaltet. Wie man hier sieht, ist das Einfügen von Text und Formatierungen problemlos möglich.