

XML, was ist das?

XML ist das standardisierte Datenformat zur Darstellung hierarchisch strukturierter Daten für den plattformübergreifenden Datenaustausch. Gerade für die elektronische Abrechnung zwischen Zahnarzt und Labor ab 2012 kommt XML zum Einsatz. Unser Autor Thomas Burgard gibt eine fundierte Einführung.

XML steht für „Extensible Markup Language“ und heißt übersetzt „Erweiterbare Auszeichnungssprache“. XML ist also keine Programmiersprache, sondern eine textbasierte Markierungssprache. XML basiert auf „SGML“ (Standard Generalized Markup Language). SGML wurde 1969 speziell für die Beschreibung von anderen Sprachen entwickelt und erst 1986 standardisiert. 1998 hat das World Wide Web Consortium (W3C) die erste Spezifikation von XML veröffentlicht. Prinzipiell gilt: XML ist ein Universal-Datenformat und kann für den plattform- und implementationsunabhängigen Datenaustausch zwischen verschiedenen Computersystemen verwendet werden. XML ist das standardisierte Datenformat zur Darstellung hierarchisch strukturierter Daten und liefert für diese Daten strukturelle und semantische Informationen. Man kann auch einfach sagen, dass XML Daten für Daten liefert.

Ein XML-Dokument besteht immer aus sogenannten ASCII-kodierten (lesbare Textzeichen) Textzeichen und besitzen die Dateiendung .xml. Binärdaten werden in einem XML-Dokument prinzipiell nicht gespeichert. Für eine Anwendungssoftware bedeutet das: Liegt ein fehlerfreies XML-Dokument vor, kann mittels beliebiger Programmiersprache eine Datenstruktur generiert und diese dann mit einem anderen Computersystem standardisiert ausgetauscht werden. XML ist somit ein sehr mächtiges Datenformat und wird deshalb auch in den meisten Softwareanwendungen zur Datenstrukturierung eingesetzt. Durch die Standardisierung von XML sind viele Applikationen und Bibliotheken für die Softwareentwicklung entstanden und erleichtert die Entwicklungsarbeit der Softwareentwickler erheblich.

Was ist XML nun genau?

1. Markierung

Wie oben bereits beschrieben, ist XML eine Markierungssprache.

Was bedeutet nun Markierung genau? Markierung besteht aus sogenannten „Tags“ bzw. „Tokens“, die dem markierten Text bzw. Textinformationen zur Veränderung des Aussehens oder der Bedeutung hinzugefügt werden. Die Tags teilen der Softwareanwendung mit, wie mit der markierten Information umgegangen wird. Eine Anwendung von XML ist die weiterentwickelte Markierungssprache des World Wide Web (WWW), der sogenannten „Extensible Hypertext Markup Language“ (X)HTML. Mit (X)HTML werden die Inhalte von Webseiten für das Internet formatiert bzw. markiert. Hier soll ein kleines Beispiel für das Verständnis dienen.

Der folgende standardisierte (X)HTML5-Code teilt einem HTML-Interpreter im Browser (z. B. Mozilla Firefox) mit, wie eine Webseite aufgebaut ist und wo die entsprechenden Inhaltsinformationen zu finden sind. Außerdem wird eine Überschrift auch als Überschrift der ersten Hierarchie markiert:

```
<html>
<head>
  <title>Das ist der Titel der Webseite</title>
</head>
<body>
  <h1>Das ist die Überschrift</h1>
</body>
</html>
```

In diesem einfachen, aber fundamentalen Beispiel kann man gut erkennen, was die Markierungstags (z. B. <h1></h1>) genau bedeuten. Zuerst ist es wichtig zu wissen, dass jede Textinformation zwischen einem Anfangs- und Endtag steht. Das ist sehr wichtig für die Anwendersoftware, denn der (X)HTML-Interpreter kann somit das Ende der markierten Information finden. Die Tags können natürlich verschachtelt werden, z. B. <head><title>Das ist der Titel der Webseite</title></head>, hier steht der Tag <title> direkt nach dem Tag <head>. Jeder Tag beginnt mit einer geöffneten spitzen Klammer „<“ und endet mit einer

geschlossenen spitzen Klammer „>“. Das Endtag beginnt immer mit den Zeichen „/>“. Dies verlangt der (X)HTML-Standard und darf vom Programmierer nicht verletzt werden, da es sonst zu Interpretationsfehlern kommt.

In diesem Beispiel werden also die Informationen so markiert, dass der (X)HTML-Interpreter die Inhaltsinformationen hierarchisch und optisch formatieren kann. (X)HTML als weiterentwickeltes HTML ist somit für die Anzeige der Informationen zuständig.

2. XML für die Beschreibung von Inhaltsinformationen

XML ist dann für die „Beschreibung“ der Inhaltsinformationen zuständig. Auch hier soll ein einführendes Beispiel dem Verständnis dienen. Das folgende Beispiel soll Kontaktadressen von Angestellten in einem Unternehmen beschreiben. Die Kontaktdaten sollen in der XML-Datei „firma-kontaktdaten.xml“ gespeichert und von einer Anwendersoftware verarbeitet werden:

```
<!-- Datei: firma-kontaktdaten.xml -->
<!-- Hier sind alle Angestellten der Firma gespeichert -->
<?xml version=„1.0“?>
<!DOCTYPE angestellte SYSTEM
„angestellte.dtd“>
<angestellte>
  <kontaktadresse>
    <nachname>Mustermann</nachname>
    <vorname>Max</vorname>
    <abteilung>Vertrieb</abteilung>
    <telefon art=„mobil“>12345</telefon>
    <email>max.mustermann@firma.de</email>
  </kontaktadresse>
  ...
  ...
</angestellte>
```

Die Anwendungssoftware kann die Datei „firma-kontaktdaten.xml“ lesen, serialisieren und über das Firmennetzwerk zu anderen Computern senden bzw. empfangen. Die Anwendungssoftware kann außerdem nach Empfang der XML-Datei diese wieder zum Lesen öffnen und mittels der dort

gespeicherten XML-Markierungen die Kontaktdaten der Angestellten ohne Probleme herauslesen und z. B. in einem Fenster anzeigen oder in einer Datenbank persistent speichern.

Prinzipiell gilt: Jede XML-Datei beginnt mit dem „Prolog“ (<?xml version=„1.0“?>), in der die XML-Version beschrieben wird und aus sogenannten „Elementen“ (Minimum ist ein Element). Die Elemente können ineinander verschachtelt werden. Groß- und Kleinschreibung ist zu beachten. Jedes Element hat einen Start-Tag und einen Ende-Tag. Start-Tags folgen der Form <NAME>, Ende-Tags der Form </NAME>, wobei NAME ein Platzhalter für einen beliebigen Typ eines Tags ist. Hier ein Beispiel für ein Element:

```
<email>max.mustermann@firma.de</email>
```

In einem besonderen Fall können die Elemente auch zusammengefasst werden. Hier ein Beispiel: <email /> und ist gleichbedeutend zu <email></email>

Kommentare werden zwischen <!-- und --> geschrieben und können beliebig oft im XML-Dokument auftauchen.

Die Zeile <!DOCTYPE angestellte SYSTEM „angestellte.dtd“> beschreibt die „Dokumententyp-Definition“ und teilt dem XML-Prozessor mit, dass das XML-Dokument nach den Regeln in der Datei „angestellte.dtd“ folgen soll. Ein XML-Dokument kann verschiedene Zustände haben:

- Ein XML-Dokument ist wohlgeformt (well-formed), wenn es die syntaktischen Regeln von XML einhält.
- Ein XML-Dokument ist gültig (valid), wenn es wohlgeformt ist und alle Regeln der Dokumententyp-Definition (DTD) einhält.

Wohlgeformte und validate XML-Dokumente

Ein „wohlgeformtes“ XML-Dokument beinhaltet prinzipiell einen fehlerfreien XML-Code, d. h. alle Elemente sind mit dem richtigen „Tag“ beendet und die korrekte Syntax ist eingehalten.

Ein „valides“ XML-Dokument benötigt in jedem Fall eine DTD. Die XML-Datei kann somit nach den Regeln der DTD auf Korrektheit geprüft werden. Ist das Ergebnis dieser Prüfung positiv, ist die XML-Datei validiert. Ein XML-Dokument besteht aus dem Prolog, genau einem Element (mit verschachtelten Elementen) und beliebig viele Kommentare, Verarbeitungsanweisungen (PIs, Processing Instructions) und Trennzeichen (space, Carriage Return, Line feed, Tab).

Was bedeutet DTD?

Die Dokumententyp-Definition, abgekürzt DTD, ist die Grundlage

für jedes XML-Dokument und definiert die exakte Struktur, Anordnung und den Inhalt. In der DTD sind also die Regeln für das XML-Dokument definiert. Des Weiteren werden in der DTD auch sogenannte Attributwerte definiert. Eine DTD ist wiederum im ASCII-Zeichensatz formatiert und kann als separate Textdatei mit der Endung „.dtd“ auch einfach gelesen und interpretiert werden. Im obigen Beispiel heißt die DTD-Datei „angestellte.dtd“.

Der Aufbau der DTD sieht wie folgt aus:

```
<!ELEMENT angestellte (kontaktadresse)
<!ELEMENT nachname (#PCDATA)
<!ELEMENT vorname (#PCDATA)
<!ELEMENT abteilung (#PCDATA)
<!ELEMENT telefon (#PCDATA)
<!ELEMENT email (#PCDATA)
<ATTLIST telefon art (mobil|fest)
#REQUIRED>
```

Attributwerte in einer XML-Datei und deren Definition in der DTD: In unserem Beispiel ist das Element <telefon> mit dem Attributwert art=„mobil“ ausgestattet: <telefon art=„mobil“>12345</telefon>

Dieses Attribut mit den möglichen Werten (mobil oder festnetz) muss ebenfalls in der DTD definiert werden:

```
<ATTLIST telefon art (mobil|festnetz) #REQUIRED>
```

In unserem Beispiel muss das Attribut „art“ mit dem Wert „mobil“ oder „festnetz“ zwingend angegeben werden.

Fazit

XML hat sich in den letzten Jahren als Markierungssprache für Dateninhalte durchgesetzt und wird gerade für den plattform- und implementationsunabhängigen Datenaustausch zwischen verschiedenen Computersystemen sehr stark eingesetzt. XML ist sehr einfach zu erlernen und für Softwareentwickler einfach zu implementieren. Es gibt viele und sehr leistungsfähige Softwarewerkzeuge auf dem Markt, mit denen die tägliche Arbeit mit komplexen Datenstrukturen um ein Vielfaches vereinfacht wird. Gerade wegen der Einfachheit und der Standardisierung wird XML in der Informatik gerne für den elektronischen Datenaustausch bevorzugt und hat sich im Prinzip durchgesetzt, ab 2012 auch in der elektronischen Abrechnung für Zahnärzte und Dentallabore. **ZT**

ZT Adresse

Thomas Burgard
Dipl.-Ing.(FH) Softwareentwicklung & Webdesign
www.burgardsoft.de



ANZEIGE

64 Bit ... mit XML-Modul

LABOR EXPRESS

Die Abrechnungs-Software für das Dental-Labor

Der Service stimmt!

CD anfordern!
kostenlos testen

BEYCODENT

Wolfsweg 34 • D-57562 Herdorf

12a02

Jetzt informieren:
Telefon: 02744 / 920837

www.dental-laborsoftware.de

Die Dokumententyp-Definition, abgekürzt DTD, ist die Grundlage