

1 Vorwort

XML-Dateien sind hierarchisch als Baum organisiert. Eine Datei hat eine Wurzel und mehrere Verzweigungen. Wie bei einem Stammbaum spricht man hier von **Eltern**, **Kindern** und **Geschwistern**.

An XML-Dateien werden unterschiedliche Anforderungen gestellt.

Die niedrigste Anforderung ist das sie **wohlgeformt** ist. Das bedeutet, das zu jedem öffnenden Tag auch ein schließendes Tag vorhanden ist. Außerdem müssen beide Tags auf der gleichen Ebene sein. Wohlgeformtheit lässt sich einfach überprüfen, in der Regel sorgt bereits der Editor dafür.

Die nächst höhere Ebene ist die **Gültigkeit** oder **Validität**. Eine valide XML-Datei muss den Regeln einer DTD oder eines Schemas folgen. Zur Überprüfung werden spezielle Programme wie z. B. xmllint benutzt. Die Validität sagt nur etwas über die Gültigkeit der **Form** aus, sie prüft nicht den Inhalt.

Die Überprüfung des Inhalts erfolgt durch einen **Parser**. Dieser kann den **Inhalt** der Elemente auf Übereinstimmung mit zulässigen Werten prüfen.

2 Aufbau

Am Beispiel einer Kurzmitteilung (note) soll hier der Aufbau einer DTD gezeigt werden. Eine note hat einen Absender (from), einen Empfänger (to) einen Betreff (heading) und einen Text (body).

2.1 Elemente

Eine DTD definiert Elemente ELEMENT und deren Eigenschaften.

```
<!ELEMENT Elementname Category>  
oder  
<!ELEMENT Elementname (Elementinhalt)>
```

2.1.1 Leere Elemente

```
<!ELEMENT Elementname EMPTY>
```

Beispiel:

```
<!ELEMENT br EMPTY>
```

XML Beispiel:

```
<br />
```

2.1.2 Elemente mit Text

<!**ELEMENT** Elementname (#PCDATA)>

Beispiel:

<!**ELEMENT** from (#PCDATA)>

2.1.3 Elemente mit beliebigem Inhalt

<!**ELEMENT** Elementname **ANY**>

Beispiel:

<!**ELEMENT** note **ANY**>

2.1.4 Elemente mit Kindern

<!**ELEMENT** Elementname (child1)>

oder

<!**ELEMENT** Elementname (child1 , child2 ,...)>

Beispiel:

<!**ELEMENT** note (to , from , heading , body)>

Die Elemente müssen in der XML-Datei in genau der Reihenfolge stehen, wie sie hier definiert sind.

Für die Kurzmitteilung werden folgende Elemente benötigt:

<!**ELEMENT** note (to , from , heading , body)>

<!**ELEMENT** to (#PCDATA)>

<!**ELEMENT** from (#PCDATA)>

<!**ELEMENT** heading (#PCDATA)>

<!**ELEMENT** body (#PCDATA)>

2.1.5 Einmalige Elemente

<!**ELEMENT** Elementname (child-name)>

Beispiel:

<!**ELEMENT** note (heading)>

Das Element heading muss **genau ein Mal** in der Datei vorkommen.

2.1.6 Ein oder mehrmalige Elemente

<!**ELEMENT** Elementname (child-name+)>

Beispiel:

<!**ELEMENT** note (to+)>

Das Element to muss **mindestens ein Mal** in der Datei vorkommen.

2.1.7 Null oder mehrmalige Elemente

<!**ELEMENT** Elementname (child-name*)>

Beispiel:

<!**ELEMENT** note (body*)>

Das Element body darf **beliebig oft** auch **null Mal** auftreten.

2.1.8 Null oder einmalige Elemente

<!**ELEMENT** Elementname (child-name?)>

Beispiel:

<!**ELEMENT** note (body?)>

Das Element **body** darf **null** oder **ein Mal** auftreten.

2.1.9 Alternative Elemente

<!**ELEMENT** note (to, from, header, (message|body))>

Die Elemente to, from und header müssen vorkommen, außerdem muss entweder ein message oder ein body Element vorhanden sein.

2.2 Attribute

Attribute sollten sparsam eingesetzt werden, da Elemente einfacher verarbeitet werden können.

3 Abkürzungen

DTD Document Type Definition (Dokumenttypdefinition)

XML Extensible Markup Language (Erweiterbare Auszeichnungssprache)

4 Lizenz

- Copyright 2019 Axel Kielhorn

- Lizenz: CC-BY-SA 4.0 Unported <http://creativecommons.org/licenses/by-sa/4.0/deed.de>