

# XML

## U8 XML

Definition .....	41
Aufgaben.....	41
Vorteile .....	41
Regeln .....	41
Element oder Attribut.....	41
XML und HTML .....	42
well formed und valid .....	42
Aufbau: .....	42
extensible.....	43

# XML

## Die standardisierte Datenfreiheit

Die Extensible Markup Language (engl. für „erweiterbare Auszeichnungssprache“), abgekürzt XML, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien. XML wird u. a. für den plattformunabhängigen Austausch von Daten zwischen Computersystemen eingesetzt, insbesondere über das Internet.

XML ist „extensible“, da es sich im Gegensatz zu HTML nicht um ein festes Format im Sinne einer bestimmten Auszeichnungssprache, sondern vielmehr um eine Metasprache handelt, die Vorschriften bereitstellt, eine beliebige Vielzahl konkreter Auszeichnungsmöglichkeiten für die verschiedensten Arten von Dokumenten zu definieren. Neben der Unterstützung des Dokumentenaustausches ermöglicht XML allgemein die flexible Weiterverwendung von Daten.

Die vom World Wide Web Consortium (W3C) herausgegebene XML-Spezifikation 1.0 definiert eine Metasprache, auf deren Regelwerk basierend durch strukturelle und inhaltliche Einschränkungen anwendungsspezifische Sprachen definiert werden. Beispiele für XML-konforme Sprachen sind: RSS, XHTML, XAML, aber auch XML-Schema.

Ein XML-Dokument besteht aus Textzeichen, im einfachsten Fall in ASCII-Kodierung (zumeist üblich ist hier die UTF-8 Kodierung) und ist damit menschenlesbar. Binärdaten enthält es per Definition nicht.

### Definition

- XML ist eine einfache, plattformunabhängige Internetsprache, die nichts macht

### Aufgaben

- speichern, sichern, ablegen, strukturieren und übertragen von Daten
- Beschreibung von Inhalten
- XML kann Inhalte nicht darstellen, hierzu benötigt man eine CSS-Datei

### Vorteile

- kein Datenverlust bei Plattformwechsel, da reine Textdatei
- kann von Lesemaschinen und Readern wiedergegeben werden
- Tags müssen nicht vordefiniert sein, es können eigene Tags erstellt werden

### Regeln

- alle Elemente müssen ein schließendes Tag haben />
- XML Tags sind case sensitive
- XML Elemente müssen richtig ineinander verschachtelt werden
- XML Attribute müssen in Anführungszeichen gesetzt werden
- Eintität richtig referenzieren
- Leerzeichen sind in XML konserviert
- Namen können Buchstaben, Zahlen und andere Zeichen beinhalten
- Namen dürfen nicht mit einer Nummer oder einem Interpunktionszeichen beginnen
- Namen dürfen nicht mit der Buchstabenkombination xml oder XML beginnen
- Namen dürfen keine Leerzeichen enthalten
- Am Anfang jeder XML-Datei muss eine XML-Deklaration stehen.
- Innerhalb eines Tags darf ein Attributname nur einmal vorkommen
- Jedes Dokument muss ein sogenannten Root-Element enthalten, das alle anderen Elemente einschließt (wie z.B. in HTML das html-Element).
- Elemente dürfen sich nicht überlappen (statt <a><b> ... </a></b> immer <a><b> ... </b></a>)
- Die Namen von Anfangs- und End-Tag müssen untereinander und mit der DTD übereinstimmen (Groß- und Kleinschreibung beachten).

### Element oder Attribut

#### Element:

- - kann mehrere Werte enthalten
- - folgt der Baumstruktur
- - einfach zu spezifizieren (für zukünftige Änderungen)
- - einfach zu lesen und zu editieren

Start-Tag: <person>

End-Tag: </person>

Jedes Start-Tag braucht ein entsprechendes End-Tag z.B.

```
<person>
<geschlecht>weiblich</geschlecht>
<vorname>Anna</vorname>
<nachname>Smith</nachname>
</person>
```

#### Attribut:

- - kann nur einen Wert enthalten
- - folgt keiner Baumstruktur
- - schwer zu spezifizieren
- - schwieriger zu lesen und zu editieren

```
<person geschlecht="weiblich">
<vorname>Anna</vorname>
<nachname>Smith</nachname>
</person>
```

#### Element mit Attribut

```
<Name Vorname="Max" Nachname="Mustermann"></Name>
Dieser Code hat ein leeres Element Name. Das Element Name hat zwei Attribute Vorname und Nachname. Attribute müssen Werte enthalten.
```

## XML und HTML

### XML

- eigene Tags/Attribute möglich
- beliebig erweiterbar
- universell einsetzbar (CMS, Datenbank, Listenverwaltung)
- enthält keine Informationen über das Layout

### HTML

- nur vordefinierte Tags/Attribute
- statischer Befehlssatz
- dient nur zur Darstellung von Informationen
- Layout und Informationen sind vermischt
- XML kann HTML nicht ersetzen.

## well formed und valid

Eine XML-Datei ist *well formed* oder *wohlgeformt*, wenn die Datei die Regeln von XML korrekt einhält. Wenn die Datei dann noch zusätzlich eine DTD (Doctype definition) enthält, wird sie als *valid* oder *gültig* bezeichnet.

### well formed

- XML-Deklaration am Beginn der Seite
- mindestens ein Datenelement
- Root Element muss vorhanden sein

### valid

- XML-Deklaration am Beginn der Seite
- mindestens ein Datenelement
- Root Element muss vorhanden sein
- Doctype definition

## Aufbau:

XML Dokumente haben eine Baumstruktur.

Sie starten mit dem Wurzelement und diversifizieren sich dann bis zu „Zweigen“ und „Blättern“.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?> XML-Deklaration
<?xml-stylesheet type="text/css" href="style.css"?> Verknüpfung CSS
<!DOCTYPE kontakt [
  <!Element kontakt (datensatz)+>
  <!Element datensatz (nachname, vorname)>
  <!Element nachname (#PCDATA)> Doctype definition
  <!Element vorname (#PCDATA)>
]>
<kontakt> Root Element (Wurzelement)
  <datensatz> Child Element
    <nachname> Zufall </nachname>
    <vorname> Rainer </vorname>
  </datensatz> Ende Child Element
  <datensatz> Child Element
    <nachname> Schweiß </nachname>
    <vorname> Axel </vorname>
  </datensatz> Ende Child Element
</kontakt>
```

XML kann HTML  
nicht ersetzen!

### extensible

XML Elemente können erweitert werden um mehr Information zu transportieren.

