

Wohlgeformtheit:

- Zu jedem Anfangs-Tag gibt es ein Ende-Tag
- Voraussetzung für die eindeutige Bestimmung hierarchischer Datenstrukturen
- Existenz leerer Tags (<kekse></kekse> oder kurz <kekse />)

Aufbau

[Prolog]

[Document Type Definition]

Wurzelement (→ Baumstruktur)

// Kommentare und Verarbeitungsanweisungen an jeder Stelle außerhalb des Markups

DTD

Elemente

Elementquantifikationen

A	genau 1x
A+	mindestens 1x
A?	0x oder 1x
A*	Mindestens 0x
-(A)	A darf nicht vorkommen.
A B	eines von beiden, aber nicht beide !
A , B	A und B müssen in dieser Reihenfolge vorkommen
A & B	A und B müssen in bel. Reihenfolge vorkommen

Datentypen

#PCDATA	Beliebiger Inhalt (eines Entities / wird geparsed)
EMPTY	leer ...
ANY	“irgendwas”, auch geparsed
(KIND1,KIND2...)	kann Elemente des Typs “Kind1” und “Kind2” enthalten

Attribute

Attributparameter

#REQUIRED	Attribut muss existieren
#FIXED	Konstante
#IMPLIED	“Abgeleitet” → optional (das Programm muss den Wert ermitteln)

Attributdatentypen

CDATA	“characters data” - eine beliebige Zeichenkette (wird nicht geparsed)
ID	das Attribut “identifiziert” dieses “Tag”
IDREF / IDREFS	verweist auf eine oder mehrere ID(s) [das zu referenzierende Element muss ein Attribut des Typs “ID” besitzen !]
NMTOKEN / NMTOKENS	Ziffern, Buchstaben und Unterzeichen (oder mehrere durch Leerzeichen getrennt)
ENTITY / ENTITIES	<!ENTITY pizza "3.14 * z ² * a">, Zugriff per &pizza

Beispiel einer XML-File:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE email [
  <!ELEMENT message(subject,text)> <!-- subject,text jeweils einmal, in dieser Reihenfolge -->
  <!ATTLIST message
    from #PCDATA #REQUIRED
    to #PCDATA #REQUIRED
  <!ELEMENT subject (#PCDATA)>
  <!ELEMENT text (#PCDATA)>
]>

<message to="you@mail.tld" from="uni@christian-eichler.de">
  <!-- Kommentar des Meisters -->
  <subject>Betreff...</subject>
  <text> Irgend ein total langweiliger Text :)
    <![CDATA[
      <hr /> Meine sonstigen Tags, die nicht geparkt werden sollen ... <hr />
    ]]>
  </text>
</message>
```

XLink

Verweise auf ganze Dokumente !

```
<PizzaListe xmlns:xlink="http://www.w3.org/1999/xlink">
  <Pizza xlink:type="simple" xlink:href = "../salami.xml" xlink:title="Pizza Salami">...</Pizza>
</PizzaListe>
```

XPointer

Zum verweisen auf Stellen innerh. eines Dokuments.

xlink:href="salami.xml#Rezept"

XPath

<pre>/AAA/BBB/ <AAA> <BBB /> <BBB /> <BBB /> <BBB /> <BBB /> </AAA></pre>	<pre>/AAA/BBB[1]/ <AAA> <BBB /> <BBB /> <BBB /> <BBB /> <BBB /> </AAA></pre>	<pre>/AAA/BBB[last()]/ <AAA> <BBB /> <BBB /> <BBB /> <BBB /> <BBB /> </AAA></pre>
<pre>//CCC <AAA> <BBB> <CCC /> </BBB> <DDD /> <CCC /> <FFF /> </AAA></pre>	<pre>/AAA/* <AAA> <BBB /> <CCC /> <DDD /> <CCC /> <FFF /> </AAA></pre>	<pre>//BBB/child::* <AAA> <BBB> <CCC /> <DDD /> <CCC /> </BBB> <FFF /> </AAA></pre>

Folie 9 (XML) S.44-56 für Genaueres ;)

Anomalien

Wie bei Tabellen kann Redundanz auftreten, was zu Anomalien führt.