DTD-Syntax

Element-Deklaration

```
<!ELEMENT elementname (inhaltsmodell)>
```

Operatoren im Inhaltsmodell

Verknüpfungen

,	gefolgt von (Sequenz von Elementen)
|	oder (Auswahl aus Elementen)

pro Gruppe nur einen von beiden Verknüpfungsoperatoren verwenden – nicht mischen!

Häufigkeit (Mengenoperatoren)

	kein Operator: muss genau einmal vorkommen	(1)
?	optional, kann einmal vorkommen	(0-1)
*	optional, kann beliebig oft vorkommen	(0-n)
+	benötigt, muss mind. einmal vorkommen	(1-n)

Gruppierung

(	Öffnet Gruppen und Inhaltsmodell
)	Schließt Gruppen und Inhaltsmodell

Textinhalt im Inhaltsmodell

Nur Text – Schlüsselwort #PCDATA

```
(#PCDATA)
```

Text und Elemente gemischt – Mixed Types

```
(#PCDATA | elem1 | elem2 | ... | elemN)*
```

Mixed Types lassen sich nur nach obigem Schema definieren. Eine festgelegte Sequenz von Elementen innerhalb von Textinhalt sind somit nicht möglich. Es muss auch keines der Elemente zwingend vorkommen. Mixed Types sollte man deshalb am besten nur für optionale Inline-Elemente verwenden.

Leere Elemente deklarieren

```
<!ELEMENT elementname EMPTY>
```

Elemente mit beliebigem Inhalt deklarieren

```
<!ELEMENT elementname ANY>
```

So deklarierte Elemente „zerstören“ die Struktur und sollten deshalb vermieden werden (Zweck: Debugging)

Attributlisten-Deklaration

```
<!ATTLIST elementname
   attributname typ einstellung >
```

Attributtypen – Schlüsselworte

CDATA	Zeichendaten
NMTOKEN	Namenstoken (Regeln für XML-Bezeichner werden angewendet)
NMTOKENS	1-n leerzeichengetrennte **NMTOKEN**
ID	eindeutiger Bezeichner (**NMTOKEN**-Regeln gelten!)
IDREF	Verweis auf ID eines anderen Elements im XML-Dokument
IDREFS	1-n leerzeichengetrennte **IDREF**

Attributtypen – Aufzählung gültiger Werte

(a|b|c)	Die einzelnen Werte dürfen keine Leerzeichen enthalten

Attributeinstellungen

"*vorgabewert*"	Voreinstellung wenn das Attribut nicht angegeben wird
#REQUIRED	benötigt, Attribut muss angegeben werden, damit Dokument validiert
#IMPLIED	optional, Attribut muss nicht angegeben werden

Entity-Deklaration

```
<!ENTITY entityname "entitywert" >
```

Entities funktionieren wie Abkürzungen / Variablen. Wird im XML-Dokument **&entityname;** geschrieben, wird bei der Weiterverarbeitung der in Anführungszeichen deklarierte Entity-Wert verwendet.

Parameter-Entity-Deklaration

```
<!ENTITY % entityname "entitywert" >
```

Parameter-Entities funktionieren wie Entities, werden aber nicht in XML-Dokumenten, sondern in DTDs verwendet. Wird in der DTD nach der Parameter-Entity-Deklaration **%entityname;** geschrieben, wird der in Anführungszeichen deklarierte Entity-Wert verarbeitet. Dadurch lassen sich Redundanzen in DTDs vermeiden.

XML-Grundlagen

Kurzreferenz

Diese Referenz ist unvollständig. Sie enthält nur die gebräuchlichsten Elemente der XML- und DTD-Syntax.

Es fehlen insbesondere:

- externe Entity- und Parameter-Entity-Deklarationen
- Ungeparste Entities und Notations mit den dazugehörenden Attributtypen ENTITY und ENTITIES
- Attributeinstellung #FIXED
- Bedingte DTD-Abschnitte mit IGNORE und INCLUDE
- Erklärung von PUBLIC-Doctypes
- Vordefinierte Attribute xml:space und xml:lang und deren Funktion

www.dokumediacoach.de

XML-Syntax

XML-Deklaration

Version der XML-Spezifikation (benötigt)

```
<?xml version="1.0" encoding="UTF-8"
 standalone="no" ?>
```

optional:

Zeichenkodierung des Dokuments

Standalone-Deklaration

no – Parsen durch externe Dateien beeinflusst

yes – Parsen nicht durch externe Dateien beeinflusst

Elemente und Attribute

In XML besteht jedes Element aus 1 Start- und 1 End-Tag:

Start-Tag (Anfangsmarke)

```
<elementname attributname="attributwert">
```

Attribut

Attributwerte immer in Anführungszeichen (doppelte oder einfache möglich)

```
</elementname>
```

End-Tag (Endemarke)

Die Elemente müssen in XML hierarchisch verschachtelt sein (statt `<a><b></a></b>` → `<a><b></b></a>`).
Es gibt immer genau ein Wurzelelement.

Leere Elemente

Leere Elemente haben keine Unterelemente und keinen Text als Inhalt.

Für leere Elemente gibt es eine abkürzende Schreibweise:

```
<leer></leer>   ⟶   <leer/>
```

www.dokumediacoach.de

Vordefinierte Entities

Zeichen mit Sonderfunktion in XML müssen im Textinhalt von Elementen und in Attributwerten maskiert werden.

Dafür gibt es folgende vordefinierte Entities:

Zeichen	Entity	Dezimal	Hexadezimal
<	<	<	<
>	>	>	>
"	"	"	"
'	'	'	'
&	&	&	&

Neben der Möglichkeit, die Zeichen in Form von Entities zu maskieren, kann man auch direkt den Dezimalwert oder Hexadezimalwert eines Zeichens angeben (Unicode-Zeichentabelle). Das geht auch mit allen anderen Buchstaben und Sonderzeichen.

CDATA-Abschnitte

```
<![CDATA[ Dieser Text wird bei der
 Verarbeitung der XML-Datei ungeparst
 wieder ausgegeben ]]>
```

CDATA-Abschnitte dürfen alle Zeichen enthalten außer 2 schließenden eckigen Klammern hintereinander: `]]`

Kommentare

```
<!-- Dieser Text wird bei der
 Verarbeitung der XML-Datei ignoriert -->
```

Kommentare dürfen alle Zeichen enthalten außer 2 Minuszeichen hintereinander: `--`

Processing Instructions

```
<?target Anweisung an einen Prozessor ?>
```

Mit **target** wird ein bestimmter Prozessor angesprochen, als Inhalt steht das, was dieser Prozessor versteht.
Vergleichbar PHP in (X)HTML: `<?php ... ?>`

DOCTYPE-Deklaration

Schlüsselwort **DOCTYPE**

Externe ID verweist auf externe Teilmenge der DTD

```
<!DOCTYPE name Ext-ID [ Deklarationen ] >
```

Dokumenttyp-Name, normalerweise Name des Wurzelelements

interne Teilmenge der DTD

öffnet interne Teilmenge

schließt interne Teilmenge

interne Teilmenge der DTD

```
<?xml version="1.0"?>
<!DOCTYPE irgendwas
[
  Deklarationen im XML-Dok.
]>
```

DOCTYPE-Deklaration enthält Deklarationen in der internen Teilmenge der DTD.

externe Teilmenge der DTD

```
<?xml version="1.0"?>
<!DOCTYPE irgendwas
  SYSTEM "irgendwas.dtd">
```

DOCTYPE-Deklaration verweist auf die externe Teilmenge der DTD.

Datei **irgendwas.dtd** enthält Deklarationen

externe und interne Teilmenge der DTD

```
<?xml version="1.0"?>
<!DOCTYPE irgendwas
  SYSTEM "irgendwas.dtd"
[
  zusätzliche Deklarationen
  im XML-Dokument
]>
```

DTD Kombination von externer + interner Teilmenge.

Datei **irgendwas.dtd** enthält Deklarationen