

XML – Was ist das und wie ist der Aufbau (kurz und knapp)

Die generelle Struktur eines XML-Dokuments unterteilt sich in die drei Teilbereiche/ Sektionen **Prolog**, **Wurzelementbereich** und **Diverses**.

Innerhalb jeder Sektion können *Markups (Tags)*, im Allgemeinen Start- und Ende-Tags sowie *Entitäten (Entitys)*, als auch *Character Data* enthalten sein.

- **Markups** : Auszeichnungen werden in XML durch das so genannte Markup-Prinzip realisiert. Dies bedeutet, dass der auszuzeichnende Text jeweils durch eine öffnende und schließende Marke umklammert ist.

Anstelle von „Marke“ wird auch häufig der Begriff „Tag“ verwendet. Der Dokumentbereich von einer öffnenden bis hin zur schließenden Marke wird als *Element* bezeichnet.

- Jedes XML-Markup muss einen öffnenden und schließenden Tag besitzen.

```
<name>Test</name>
```

- Groß- und Kleinschreibung wird strikt unterschieden.

```
<Name>Test</name>
```

Das obige Beispiel wäre falsch, da es kein Äquivalent zum öffnenden Tag `Name` gibt.

- Innerhalb eines XML-Files existiert genau ein Wurzelement.

```
<message>
```

- Leerraum zwischen den öffnenden Klammern und den Markupnamen ist verboten. Ein Leerraum zwischen vor dem schließenden Tag ist dagegen erlaubt.

```
< name>Test test</name >
```

Der öffnende Tag ist fehlerhaft, der schließende ist richtig.

- Der Name eines Tags kann relativ freizügig vergeben werden. Es muss sich an folgende Grenzen gehalten werden.

- Das erste Zeichen muss Unicode - Zeichensatz (UCS) sein, oder ...
- ein Unterstrich (_), oder ...
- ein Doppelpunkt (:).

Ab dem zweiten Zeichen können andere Zeichen, ein Punkt (.) oder Minus (-) genutzt werden.

- **Entität** : Ein Entität (oder Entity) übernimmt die Funktion eines Textbausteins, d.h. die im Dokument vorhandene Zeichenkette wird durch einen Text ersetzt, der ansonsten nicht oder nur umständlich dargestellt werden könnte.

Ein Beispiel: Das bekannte „Kaufmanns-Und“ (&), wird sowohl in HTML/ XHTML als auch in XML durch die Entität & dargestellt.

- **Character Data** : Als Character Data wird alles bezeichnet, was nicht in die Bereiche Markup oder Entität gehört – also Text/ Inhalt.
- **Wurzelement**: Dieses enthält die eigentlichen Daten und besteht aus Character Data, welche durch Marken geklammert ist.

```
<name>Test</name>
```

- Diese Elemente können verschachtelt werden. Hier ist auf die genaue Reihenfolge zu achten. Damit das ganze leserlich bleibt, werden die Elemente durch TAB's oder Leerzeichen übersichtlich dargestellt.

```
<inhalt>
  <zutat>Wasser</zutat>
  <menge>1cl</menge>
</inhalt>
```

- Unterschieden wird allerdings beim *leeren* Element, also ein Element ohne Inhalt. Hierfür existieren zwei verschiedene Darstellungen.

```
<zutat>Sauerstoff</zutat>
  <menge></menge>
```

```
<zutat>Sauerstoff</zutat>
  <menge />
```

- Ebenfalls ist es möglich Elemente mit Character Data zu mischen, was aber möglichst vermeiden werden soll, da die Lesbarkeit des XML - Files gesenkt wird.

```
<inhalt>
  Auch hier können Character Data hinterlegt werden ...
  <zutat>Wasser</zutat>
  <menge>1cl</menge>
</inhalt>
```

- **Kommentare**: In XML- Dokumenten können Kommentare hinterlegt werden, die allerdings auch mit Entitys geschrieben werden müssen.

```
<!-- Kommentar können nicht mit Umlauten geschrieben werden -->
<!-- Besser ist es so: Kommentare koennen nicht ... -->
```

- **Attribute:** Wie in vielen Sprachen, besteht auch bei XML die Möglichkeit die Tags mit Hilfe von Attributen zu erweitern. In folgender Zeile wird der Tag `name` mit dem Attribut `klasse` erweitert.

```
<name klasse='Elementgruppe'>Wasser</name>
```

- Attribute dürfen nur in öffnenden Tags genutzt werden.
- Die Reihenfolge ist beliebig, insofern mehrere Attribute gesetzt werden.
- Attribute müssen Werte enthalten.
- Attributwerte werden durch Quotes ("") oder Apostrophe geklammert (' ')
- Quotes und Apostrophe können geschachtelt werden.
- Die mehrfache Angabe von Attributen ist nicht erlaubt.

```
<name klasse='Elementgruppe' klasse='Element'>Wasser</name>
```

- Das Zeichen „<“ ist im Attributwert verboten. Es kann nur in Verwendung mit Entitys genutzt werden. (<). Das gleiche gilt für andere Sonderzeichen.

```
<!-- falsch -->
```

```
<name klasse='Elementgruppe & Element'>Wasser</name>
```

```
<!-- richtig -->
```

```
<name klasse='Elementgruppe & Element'>Wasser</name>
```

Beispiel-Code

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE message SYSTEM "Adresse oder .dtd">
<!-- Willkommen in der XML-Message -->
<message>
  <product>
    <name>Test</name>
    <inhalt>
      <zutat klasse='Elementgruppe'>Wasser</zutat>
      <menge>1cl</menge>
    </inhalt>
    <inhalt>
      <zutat>Feuer</zutat>
      <menge>2ml</menge>
    </inhalt>
    <inhalt>
      <zutat>Erde</zutat>
      <menge>3dl</menge>
    </inhalt>
    <inhalt>
      <zutat>Sauerstoff</zutat>
      <menge />
    </inhalt>
    <anleitung>Produktdefinition ...
  </anleitung>
</ product >
</message>
<!-- Innerhalb der <message> koennen weitere <product> folgen -->
```
