



## XML (Webservices)

---

XML ist wie HTML eine "Markup"-Language ( daher das "ML"). Markups sind Steuerzeichen (auch Tags genannt : Vorsicht, nicht mit OPC-Tags verwechseln !), die bestimmen was die Daten bedeuten, oder wie sie darzustellen sind. Z.b. bedeutet : `<h1> hallo </h1>` in HTML, daß das "hallo" sehr groß auf dem Bildschirm dargestellt wird.

HTML ist nun eine sehr genau definierte Sprache (keine Programmiersprache !!!) , die exakt festlegt, wie Informationen in einem Webbrowser dargestellt werden. Weil eine chinesische Webseite auf einem Browser in Obermenzing richtig dargestellt werden soll, ist diese allgemeine Festlegung nötig.

In Industriekommunikationen ist das aber nicht nützlich. Man möchte eigene Formate benutzen. So möchte z.b. ein Reifenhersteller die Tragfähigkeit seiner Reifen kommunizieren, und dafür das Tag "LI" (für Lastindex) verwenden. Er benutzt XML, die (eXtensible Markup Language), in der die Tags selber festgelegt werden können : `<LI> 80 </LI>`

Der Empfänger der Nachricht kennt die Tag-Definitionen des Reifenherstellers, und versteht also den Code.

## XML - Konventionen

Wenn auch die Tags selber definiert werden können, so gibt es doch ein paar Regeln, die ein XML-Dokument erfüllen muß. (Wenn es das tut, nennt man es "well formed".)

Jedes Dokument besitzt genau ein Wurzelement. Als Wurzelement wird dabei das jeweils äußerste Element bezeichnet. Im Beispiel hier "Chargenauftrag" :

Dazu kommt mindestens ein (Daten-) Element. Diese besitzen ein Beginn- und ein Ende-Tag : <Teil1> und </Teil1>

Als Dokumentkopf wird angegeben, daß es ein XML-Dokument ist : <?xml version="1.0" encoding="UTF-8" ?>

Die Version 1.0 ist die aktuelle, UTF-8 ist ein üblicher Zeichensatz

```
<?xml version="1.0" encoding="UTF-8" ?>
  <Chargenauftrag>
    <Turm1>
      <Teil1>1</Teil1>
      <Teil2>2</Teil2>
    </Turm1>
  </Chargenauftrag>
```

Ein wenig komplizierter wird das noch, wenn man zu den Elementen noch die "Attribute" hinzunimmt. Das sind weitere mögliche Werte, die ein Element "erklären" :

```
<?xml version="1.0" encoding="UTF-8" ?>
  <Chargenauftrag>
    <Turm1>
      <Teil1 Lage="unten">1</Teil1>
      <Teil2 Lage="mitte">2</Teil2>
    </Turm1>
  </Chargenauftrag>
```

Attribute sind stets in 'Hochkomma' oder "Anführungszeichen" anzugeben.

## Json

Neben speziellen komplexeren Funktionen im Umfeld von Java (daher kommt der Name : Java Script Object Notation ) wird Json zunehmend als schlankeres Datenformat zur Übertragung industrieller Daten alternativ zu XML genutzt. Für praktisch alle Programmiersprachen gibt es, wie bei XML auch, Tools zur Codierung und Decodierung.

Beispiel :

```
{
  "chargenauftrag": "Nummer 1",
  "bauteile":
  [
    {
      "name": "saugventil",
      "schlauchdurchmesser": 8,
      "maximaldruck": 16,
      "einzelteile":
      [
        "ventilkörper",
        "elektroaufsatz",
        "kabelklemme"
      ]
    },
    {
      "name": "absperrhahn",
      "schlauchdurchmesser": 8,
      "maximaldruck": 16
    }
  ]
}
```

Es geht hier nur darum, daß sie verstanden haben, daß Json eine Datencodierung darstellt, ähnlich wie XML. Die Syntax ist Nebensache, wir benutzen im Weiteren ausschließlich XML.