



# XML - Eine Einführung





- ⇒ **Extensible Markup Language, oder kurz "XML", ist eine (die !) Technologie für Web Applikationen und Services.**
- ⇒ **XML ist ein World Wide Web Konsortium Standard welcher es erlaubt eigene "Tags" bzw. XML Dokumente zu erstellen.**



## Warum brauchen wir XML - warum nehmen wir nicht einfach HTML

```
010010100101001001001010110
11101010010101010100110
00110101010101001010100
0110101010101010101010101010
1110101010101010101010101010
010010110101001010010110
0101010110101010101010101010
111010101010101010101101101
010010110101001010010110
```

```
<p><b>Mrs. Mary McGoon</b>
<br>
1401 Main Street
<br>
Anytown, NC 34829</p>
```

- ⇒ **Warum braucht es XML wenn heute jeder Browser HTML unterstützt ?**
- ⇒ **Warum XML, wenn ich im HTML doch einfach Produktkataloge, Preislisten, Dokumentationen etc. zur Verfügung stellen kann ?**
- ⇒ **Um diese Frage zu beantworten, beachte man das Beispiel eines HTML-Codes.**
- ⇒ **HTML-Tags sind für Browsing; d. h. sie dienen der Kommunikation zwischen den Endanwendern und dem Computer.**



# Aufbau von HTML

```
010010100101001001010110  
111010100101010101001101  
00110101010101001010100  
<p><b>Mrs. Mary McGoon</b>  
<br>  
1401 Main Street  
<br>  
Anytown, NC 34829</p>  
  
Mrs. Mary McGoon  
1401 Main Street  
Anytown, NC 34829
```

- ⇒ **Aufbereitet sieht das vorherige Beispiel - wie links dargestellt - aus.**
- ⇒ **HTML Tags Beschreiben also wie etwas dargestellt werden soll.**
- ⇒ **Sie beinhalten keinerlei Informationen über die Daten sondern nur wie diese dargestellt werden sollen.**



## XML Code Beispiel

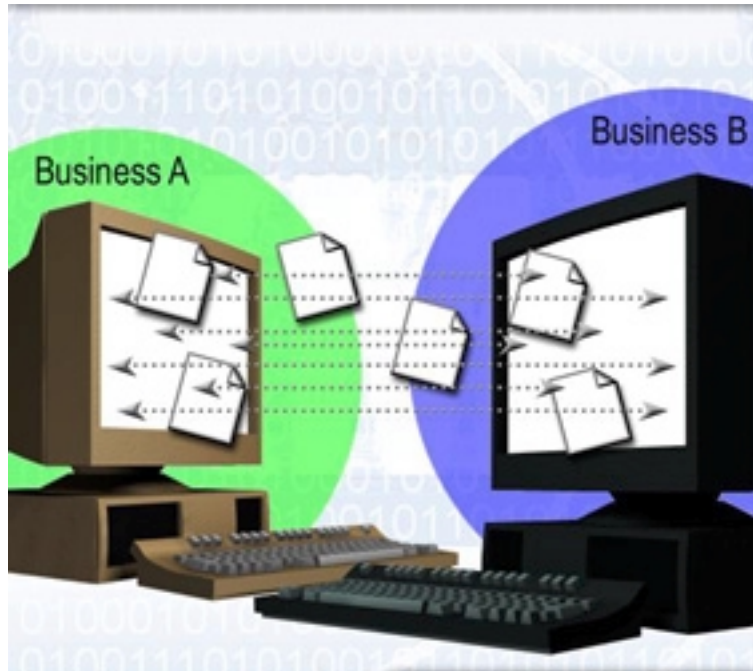
```
010010100101001001010110
111010100101010101001101
001101010101001010100
<address>
<name>
<title>Mrs.</title>
<first-name>Mary</first-name>
<last-name>McGoon</last-name>
</name>
<street>1401 Main Street</street>
<city>Anytown</city>
<state>NC</state>
<zipcode>34829</zipcode>
...
</address>
010010110101001010010110
```

- ⇒ Das gleiche Beispiel in XML Code. Mit XML, können wir die Bedeutung der Tags verstehen.
- ⇒ Noch wichtiger - der Computer kann die Bedeutung verstehen bzw. interpretieren.
- ⇒ So ist einfach für den Computer zu verstehen, dass:

`<zipcode>34829</zipcode>`

eine Postleitzahl beinhaltet.

# 1. Vorteil: Datenaustausch



## ⇒ XML wichtigster Vorteil ist die Fähigkeit zum Datenaustausch

⇒ Das ist wichtig, weil verschiedene Firmen (oder sogar Abteilungen einer Firma) selten mit den gleichen Anwendungen arbeiten und so die gegenseitige Kommunikation erschweren.

⇒ XML macht es leicht, Daten elektronisch auszutauschen, **ohne** das Informationen verloren gehen.

## ⇒ Beispiele:

⇒ WEB-Services (SOAP)

⇒ OASIS Business Transactions TC (<http://www.oasis-open.org/>)

⇒ eCH (eGovernment Initiative)

# Ein Beispiel XML Dokument

```
<?xmlversion="1.0"?>
<!DOCTYPE NewsArticle SYSTEM "News.dtd">
<NewsArticle>
<Head>
<Title>IBM Releases Open Software to Improve
Security, Performance, and Reliability of
Internet E-Mail Systems</Title>
<Date>12/14/98</Date>
<Summary>IBM is making available through
alphaWorks free open source software
designed to improve the security,
reliability and performance of
e-mail delivery services.</Summary>
<Category topic="Corporate And Financial"/>
...
</NewsArticle>
```



- ⇒ Unser Beispiel ist ein News Artikel welcher einen Titel, ein Datum, einen Text (summary) und weitere Informationen beinhaltet.
- ⇒ Welche Tags darin vorkommen und welche Reihenfolge, Verschachtelung und Anzahl wird mittels eines DTD (Document Type Definition) oder via XML Schema festgelegt. Auch können zu jedem Tag zusätzliche Attribute vereinbart werden.
- ⇒ DTD und XML stehen zueinander wie eine Klasse zu seinen Instanzen.

## 2. Vorteil: Speicherung komplexer Daten

### Beispiel: Docbook



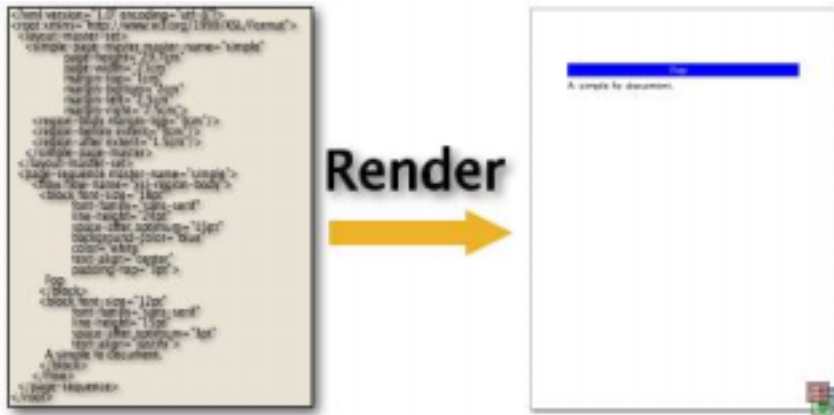
⇒ In XML können komplexe Daten gespeichert werden.

- ⇒ XML erlaubt die Speicherung hierarchischer Daten.
- ⇒ Der Inhalt der ist Daten bekannt (Sie bleiben Eigentümer der Daten !)
- ⇒ Die Korrektheit der Daten kann jederzeit mittels DTD geprüft werden.
- ⇒ Mittels Namespaces können andere Dokumente eingebettet werden.

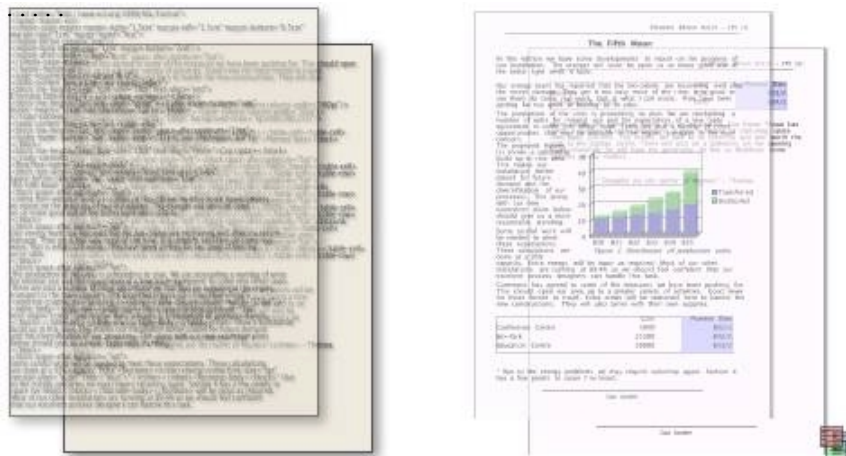
⇒ **Beispiele:**

- ⇒ DocBook ([www.docbook.org](http://www.docbook.org))
- ⇒ OpenOffice ([www.openoffice.org](http://www.openoffice.org))
- ⇒ SCODi4P ([www.scodi4p.com](http://www.scodi4p.com))

### 3. Vorteil: Anzeigen, Umwandeln, Selektionieren



```
<xsl:stylesheet xmlns:xsl="http://.....">
  <xsl:template match="/">
    <html>
      <head>
        <title><xsl:value-of select="/rows/object/title"/></title>
```



⇒ Mittels CSS können XML Dokumente im Browser dargestellt werden.

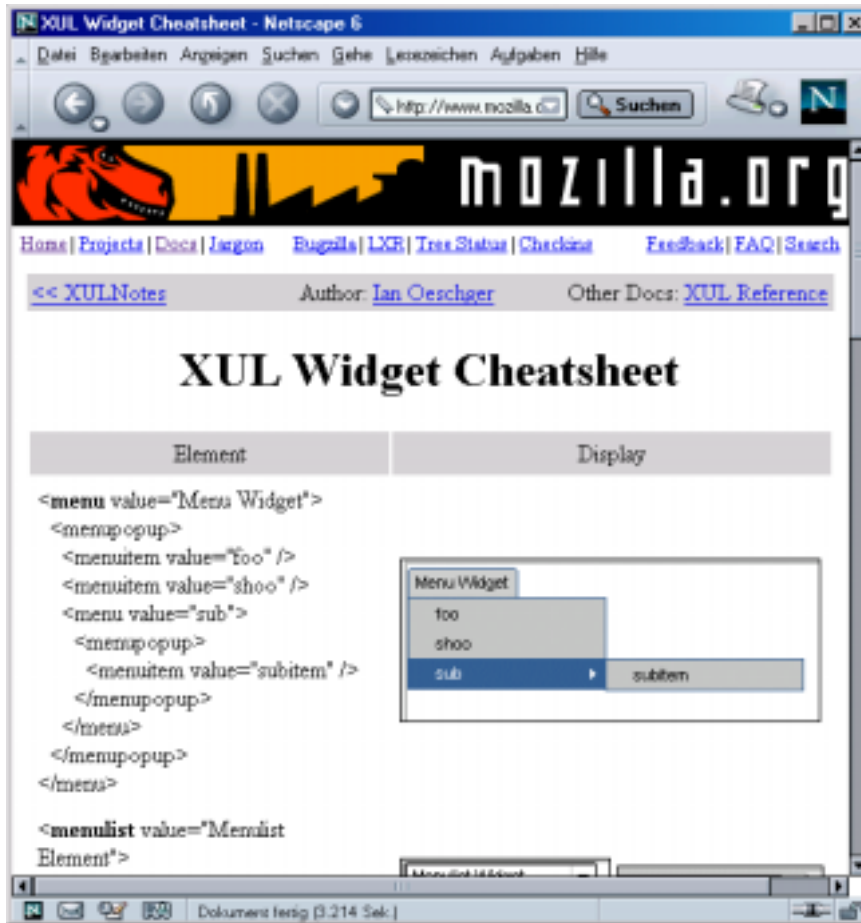
⇒ Mittels XSL können XML Dokumente in andere Formate umgewandelt werden.

⇒ XSL (XPath) bietet zusätzlich die Möglichkeit, XML Dokumente zu selektionieren (ähnlich SQL)

⇒ Beispiele:

- ⇒ Cocoon, FOP, Xalan (xml.apache.org)
- ⇒ Java ab Version 1.4
- ⇒ Wyona (wyona.org)
- ⇒ SVG (<http://xml.apache.org/batik>)
- ⇒ SCODi4P ([www.scodi4p.com](http://www.scodi4p.com))

## 4. Vorteil: Entschlacken von Applikationen



⇒ XML eignet sich vorzüglich zur Darstellung von:

- ⇒ User Interfaces
- ⇒ Listen- und Reportsprache
- ⇒ Datenbankbeschreibung (DDL)
- ⇒ Speicherung von Parametern, Optionen und Einstellungen
- ⇒ .....

⇒ Beispiele:

- ⇒ Mozilla (Gecko-Engine - XUL)
- ⇒ HISoft (<http://www.hisoft.ch>)
- ⇒ SCODi4P (<http://www.scodi4p.com>)



## Literatur & Links

### ⇒ **Standard's**

⇒ <http://www.w3c.org>, <http://www.oasis-open.org>, <http://www.docbook.org>

### ⇒ **Produkte**

⇒ <http://xml.apache.org>, <http://www.xmetal.com>, <http://www.xmlmind.com>,  
<http://www.scodi4p.com>

### ⇒ **Zeitschriften**

⇒ <http://www.sys-con.com/xml/>

### ⇒ **Tutorial's**

⇒ <http://irb-www.informatik.uni-dortmund.de/~sm/aw/xml/msie/>

### ⇒ **Bücher**

⇒ Java and XSLT, O'Reilly Verlag, Eric M. Burke, ISBN: 0-596-00143-6

⇒ XML, bhv Verlag, Thomas Kobert, ISBN: 3-8287-5044-3