

# Einführung XML

## Beginnen wir mit HTML ....

- HTML (Hyper Text Markup Language) wird zur Beschreibung von Hypertextdokumenten im WWW verwendet
- Basiskonzept: Auszeichnungen (“Markups”) in Form von “Tags”
- vom W3C (WorldWideWeb Consortium) standardisiert
- weist eine Reihe von Einschränkungen auf:
  - eine beschränkte Anzahl vordefinierten Tags
  - daher ständige Erweiterungen um (properitäre) Tags

## Beginnen wir mit HTML ....

- Tags vermischen Layout-Aspekte mit Struktur- und Inhaltsaspekten
- daher wird die Suche im Web erschwert

“The problem with what you see is what you get  
is that what you see is all you’ve got”

*Brian Kerningham*

## Von HTML zu XML

HTML beschreibt  
Layout d. Inhalts

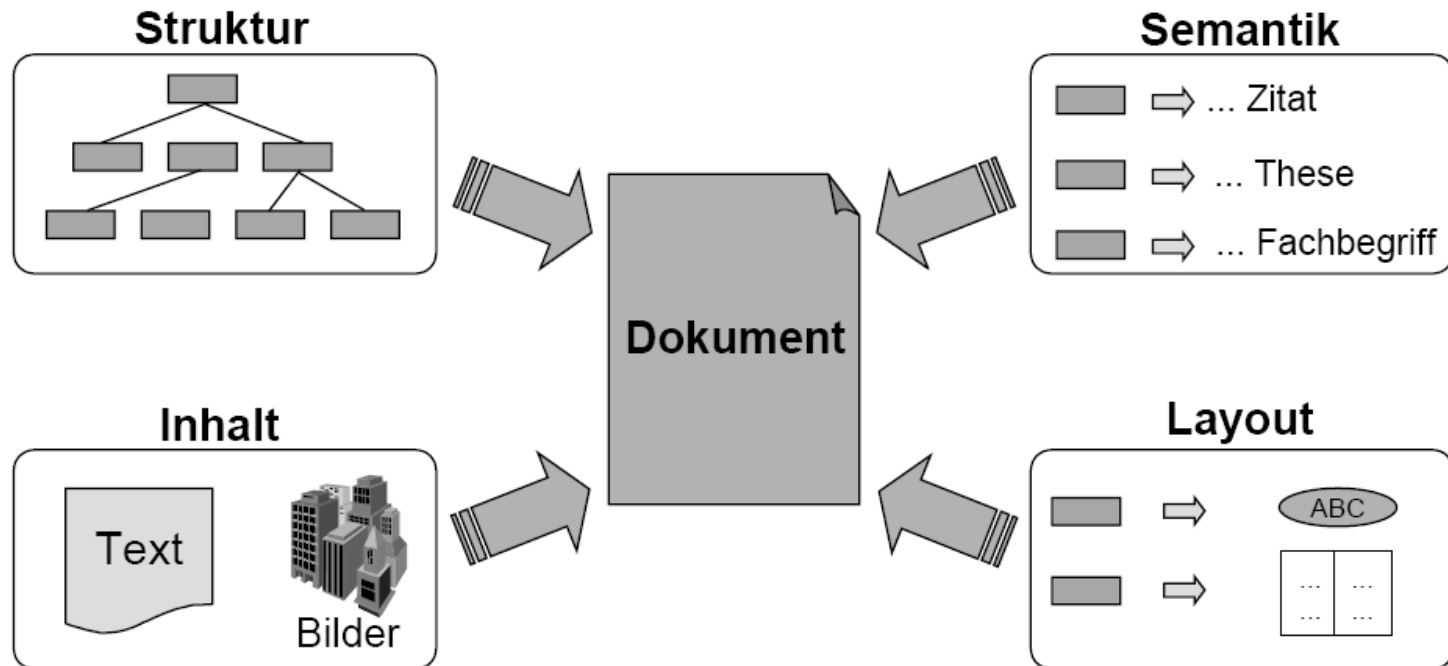
```
<h1>HandyKatalog</h1>
<h2>Nokia 8210</h2>
<table border="1">
  <tr>
    <td>Batterie</td><td>900mAh</td>
  </tr>
  <tr>
    <td>Gewicht</td><td>141g</td>
  </tr> ...
</table>
```



XML beschreibt Struktur  
und Semantik d. Inhalts

```
<HandyKatalog>
  <Hersteller name="Nokia">
    <Modell name="8210">
      <Batterie>900mAh</Batterie>
      <Gewicht>141g</Gewicht>
      ...
    </Modell>
  </Hersteller>
</HandyKatalog>
```

## Bestandteile von Dokumenten



## Beschreibung von Dokumenten

- **Visuelle Auszeichnung**

Fokus liegt in der Festlegung des **Aussehens** eines Dokuments (WYSIWYG-Prinzip)

**Strukturinformation** und **Semantik** des Textes gehen **verloren**

Beispiel: HTML

- **Generische Auszeichnung**

Autor muss sich bei Dokumenterstellung **nicht um** dessen **Formatierung kümmern**

Fokus liegt in der **Struktur des Dokuments** und der **Semantik der einzelnen Elemente**

Beispiel: XML

## Entwurfsziele von XML

- XML soll sich **im Internet** auf **einfache Weise** nutzen lassen
- XML soll ein **breites Spektrum von Anwendungsbereichen** unterstützen
- **XML-verarbeitende Anwendungen** sollen **einfach entwickelt** werden können
- Die **Anzahl optionaler Konzepte** soll **minimal** sein
- XML-Dokumente sollen **für Menschen lesbar, verständlich** und **einfach zu erstellen** sein
- Der **XML-Entwurf** soll **formal** und **präzise** sein

## Merkmale von XML

- **Erweiterbarkeit** (Metasprache)
  - Tags und Attribute können neu definiert und benannt werden
- **Strukturierbarkeit**
  - Tags können beliebig komplex geschachtelt werden
- **Selbstbeschreibendes Format**
  - Tags im XML Dokument beschreiben Struktur und Semantik des Inhalts
  - ... für den Menschen: einfach zu lesen u. zu erstellen
  - ... für die Maschine: einfach zu generieren u. zu parsen



## Merkmale von XML

- **Validierbarkeit**

XML Dokumente können optional eine formale Beschreibung ihres Vokabulars und ihrer Grammatik aufweisen (Document Type Definition - DTD bzw. XML Schema) und gegenüber dieser validiert werden

- **Layoutunabhängigkeit**

Trennung der Struktur und Semantik des Inhalts von dessen Layout

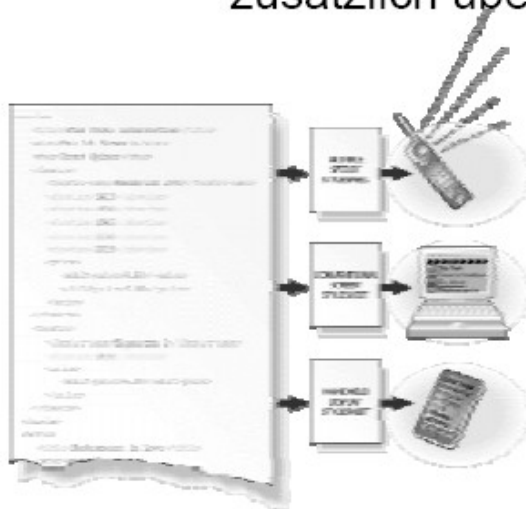
- **Semistrukturiertheit**

Inhalt kann auch nicht-strukturierte Teile aufweisen

## Anwendungsbereiche von XML

### ■ Datenaustausch

- ◆ über XML als reine Notation oder zusätzlich über gemeinsame DTDs



### ■ Multi-Delivery

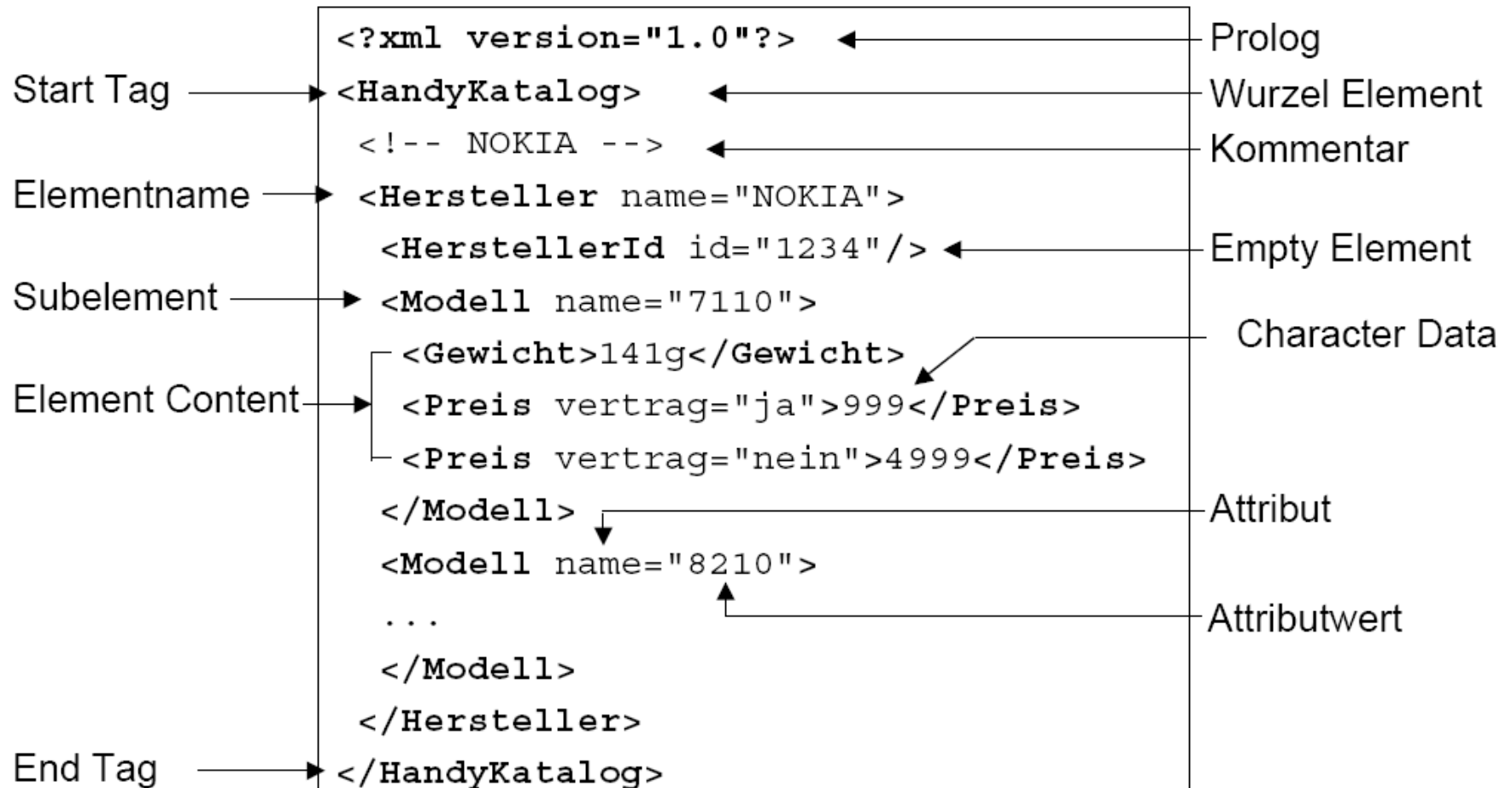
- ◆ ein und derselbe Inhalt kann auf verschiedenen Endgeräten unterschiedlich präsentiert werden

### ■ Intelligente Suche

- ◆ statt einfacher Schlagwortsuche in HTML-Dokumenten, strukturbasierte Suche in XML-Dokumenten möglich

“Mozart” -  
Komponist oder Kugel?

## XML Dokument



## XML Elemente & Attribute

- Header

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

- jedes Element muss mit einem Tag abgeschlossen werden
- Elemente dürfen sich nicht überlappen
- Leere Elemente können in Kurzform angeschrieben werden

```
<HerstellerId id="1234"></HerstellerId> oder  
<HerstellerId id="1234"/>
```

## XML Elemente & Attribute

- Attributwerte müssen unter Anführungszeichen gesetzt werden

`<Modell name='8210'>` oder

`<Modell name="8210">`

- jeder Attributname darf nur einmal pro Element vorkommen
- Namenskonventionen für Element- und Attributnamen:

`[ letter | _ | : ] [ letter | '0..9' | '.' | '-' | '_' | ':' ]*`

Case-sensitiv

- genau ein Root-Element pro Dokument

## XML Kommentare

- ein Kommentar beginnt mit der Zeichenfolge `<!--` und endet mit `-->`.

`<!-- Das ist eine Kommentarzeile. -->`

- können sich über mehrere Zeilen erstrecken
- auch zwischen Start Tags und End Tags von Elementen
- dürfen keine "--" enthalten
- ein XML-Prozessor muß den Kommentar nicht an die Anwendung weitergeben

```
<!--  
Ein Kommentar darf  
auch Dinge wie <tagnamen>  
oder &entitäten; enthalten  
-->  
...
```

## Eigenschaften von XML Dokumenten

### ■ wohlgeformt (well-formedness)

wird definiert durch bestimmte syntaktische Eigenschaften eines XML-Dokuments, z.B.:

- XML Dokument enthält ein oder mehrere Elemente
- exakt **ein** Element als Wurzel - ("Document Entity")
- keine Überlappungen bei Tags
- jedes direkt oder indirekt referenzierte analysierte Entity ist wiederum wohlgeformt

### ■ gültig (validity)

- XML Dokument ist wohlgeformt und
- entspricht einer DTD

## PCDATA vs. CDATA

- **PCDATA - Parsable Character Data**

Ein Element kann Text (Character Content) enthalten, der von einem XML-Prozessor interpretiert wird

Beispiel: `<Gewicht>141g</Gewicht>`

- **CDATA - Character Data**

Ein Element kann Text enthalten, der vom XML-Prozessor ignoriert wird, d.h. evt. Tags werden darin nicht erkannt

Beispiel: `<![CDATA[<Gewicht>141g</Gewicht>]]>`



## PCDATA vs. CDATA

- **PCDATA - Parsable Character Data**

Ein Element kann Text (Character Content) enthalten, der von einem XML-Prozessor interpretiert wird

Beispiel: `<Gewicht>141g</Gewicht>`

- **CDATA - Character Data**

Ein Element kann Text enthalten, der vom XML-Prozessor ignoriert wird, d.h. evt. Tags werden darin nicht erkannt

Beispiel: `<![CDATA[<Gewicht>141g</Gewicht>]]>`

## Aufgabenstellung

- Vorbereitung fuer den Laborunterricht, lesen Sie folgenden Tutorials durch:
  - o XML:  
<http://www.w3schools.com/xml>
  - o XML Schema:  
<http://www.w3schools.com/schema>
  - o XPath:  
[http://www.w3schools.com/xsl/xpath\\_intro.asp](http://www.w3schools.com/xsl/xpath_intro.asp)
  - o XSL Transformation:  
<http://www.w3schools.com/xsl/default.asp>

# XML Schema

## *Definition von XML Dokumenten*

## Ziel von XML Schema

- XML basierende Sprache
- Erstellung und Prüfung von XML basierenden Sprachen
- Erstellung von Datenmodellen

*„XML schema **defines element and attribute names** for a class of XML documents. The schema also specifies the **structure** that those documents must adhere to and the **type of content** that each element can hold.“*

Quelle: <http://www.learn-xml-schema-tutorial.com>

## Instanzen und Validierung

Begriffe ...

- Instance
- Valid Instance
- Well Formed XML Document
- Valid XML Document

*„An XML document can be **well formed** without being **valid**, but it cannot be valid unless it is well formed.“*

Quelle: <http://www.learn-xml-schema-tutorial.com>