



XML-Technologien

Tutorium 1

Themen:

Grundlagen XML
Syntax, Wohlgeformtheit
XML-Deklaration
Namensräume

von:

Kain Kordian Gontarska
kainkordian@gmail.com

Grundlagen XML – SGML

- XML ist ein SGML „Dialekt“ (Teilmenge)
- SGML - Standard Generalized Markup Language:
 - Motivation: Plattform- und Anwendungsunabhängiges Format zum Speichern von Informationen.
 - Bringt natürliche Sprache in eine „maschinenlesbare“ Form durch Markierung von Elementen.
 - Für Gliederung, Metainformationen und Formatierung.
 - Durchgesetzt 1986, als ISO-Norm 8879

Grundlagen XML - SGML

- SGML gibt keine konkreten Tags vor, nur eine Syntax (ua. DTD)
- Mit Hilfe der DTDs lassen sich konkrete MLs (Anwendungen) definieren
- Bekannteste Anwendungen: HTML, XML
- SGML war nicht sehr erfolgreich, da es äußerst komplex ist (155 Seiten Spezifikation) und zu viele Freiheiten bietet.

Grundlagen XML - HTML

- Auswahl an Tags ist vordefiniert, alle haben spezielle Bedeutung
- Viele Tags dienen zur Formatierung des Textes
- + sehr simpel, hat sich weiterentwickelt und bewährt zur Präsentation von Web-Inhalten
 - eignet sich nicht für die „intelligente“ Strukturierung von Inhalten

In HTML:

```
<p>Apple Titanium Notebook  
<br>Local Computer Store  
<br>$1438
```

In XML:

```
<product>  
  <model>Apple Titanium Notebook</model>  
  <dealer>Local Computer Store</dealer>  
  <price>$1438</price>  
</product>
```

Grundlagen XML

- Konsequente Vereinfachung von SGML
- Striktere Regeln → Leichtgewichtiger, einfacher annehmbar
- Ähnlich SGML – abstraktes System, welches für die Benutzung sinnvoll konkretisiert werden sollte (Einschränkung der Sprache über DTD/XML-Schema)
- Ansammlung von Regeln zur Syntax
- Baumartige Verschachtelung von Tags
- Wann Wohlgeformt?

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <element1>I am element1!</element1>
  <element2 feature="not very talkative" />
  <element1>No, I am element1!</element1>
  <somethingElse feature="not part of it all">
    <specialTag>meh.</specialTag>
  </sometingElse>
</root>
```

Wohlgeformtheit

- Elemente können aus unterschiedlichen Arten von Inhalt bestehen:
 - unstrukturierter Inhalt
 - strukturierter Inhalt
 - gemischter Inhalt
 - leerer Inhalt
- Namensvergabe für Elemente ist beschränkt auf:
 - Sie beginnen mit einem Buchstaben (groß oder klein) oder einem Unterstrich
 - Sie beginnen nie mit den Buchstaben xml (groß oder klein)
 - Enthalten nur Buchstaben, Zahlen, Punkte oder UnterstricheKeine Leerzeichen (Name endet mit Leerzeichen) oder Doppelpunkte (Reserviert für Namensraumzuordnung)

Wohlgeformtheit

- Ein XML-Dokument ist genau dann „wohlgeformt“, wenn es alle XML-Regeln einhält:
 - 1 Das Dokument besitzt genau ein Wurzelement
 - 2 Alle Elemente besitzen ein Beginn- und End-Tag. Inhaltlose Elemente können selbstschließend sein.
 - 3 Beginn- und End-Tag müssen ebentreu-paarig sein, sie dürfen nicht überlappen.
`<element><stuff></element></stuff>` ← **Falsch!**
 - 4 Eine Element darf nicht mehrere gleichnamige Attribute besitzen.
 - 5 Elementnamen müssen Namenskonvention erfüllen.
 - 6 Die Beginn- und End-Tags sind case sensitive.
 - 7 XML belässt White Space im Text. (Regel?)

XML-Deklaration

- Optionale Erkennungszeichenfolge im Prolog eines XML-Dokuments der Form `<?xml 'Attribute' ?>`
- `version="1.0|1.1"`
 - Definiert die Versionsnummer der zugrundeliegenden XML-Spezifikation. Angabe erforderlich.
- `encoding="Zeichenkodierung"`
 - Bestimmt Kodierung des XML-Dokuments. Bei Weglassen wird der Wert `encoding="UTF-8"` gesetzt.
- `standalone="yes|no"`
 - Angabe optional. Bei weglassen wird Wert `standalone="no"` gesetzt.
 - Auf "yes" setzen, falls ein internes DTD verwendet wird.
- Bei Weglassen der Deklaration, wird von der Version 1.0 ausgegangen.
- Attribute haben feste Reihenfolge (wie angegeben)

```
<?xml version="1.1" encoding="UTF-16" standalone="yes" ?>
```


Attribute

- Erweitern den Informationsgehalt von Elementen
- Jedes Kind-Element mit unstrukturiertem Inhalt lässt sich auch als Attribut darstellen (und andersrum)
- Reihenfolge der Attribute ist nicht relevant, die von Elementen ist relevant

```
<name>
  <id>12345</id>
  <first>John</first>
  <middle>Fitzgerald</middle>
  <last>Doe</last>
</name>
```

```
<name id="12345"
  first="John"
  middle="Fitzgerald"
  last="Doe" />
```

XML-Namensräume

- Wirkt der Mehrdeutigkeit eines Elementnamens entgegen.
Bsp: Titel einer Person und Titel des von ihr verfassten Buches
- Präfix gibt den Kontext an.
- Präfix-Namensraum & Standardnamensraum

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

```
<table xmlns="Irgendeine URI, am Besten zum
Thema">
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

```
<f:table
  xmlns:f="http://en.wikipedia.org/wiki/Furniture">
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```