



XML 1

Einführung, oXygen
Ulrike Henny
ulrike.henny@uni-koeln.de



XML – eXtensible Markup Language

- Was ist XML?
- XML-Grundlagen
- XML-Editoren, oXygen
- HTML + CSS
- XPath
- Übungen
- Literatur

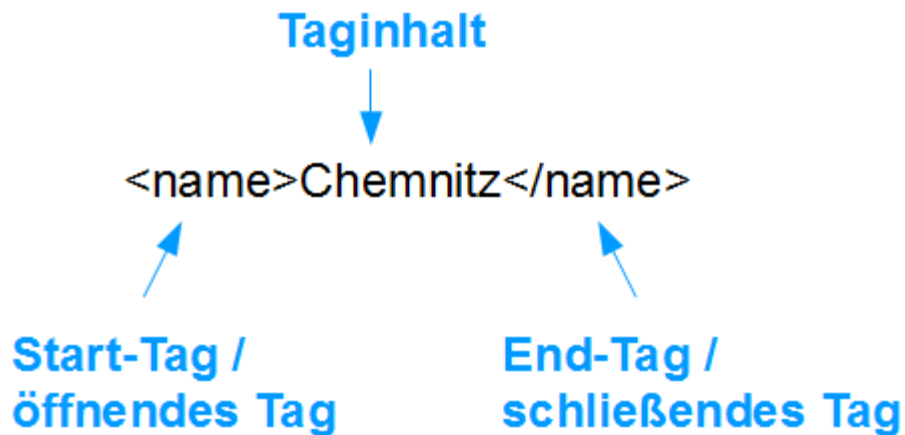
Was ist XML?

- eXtensible Markup Language
 - erweiterbar:
 - allgemeine Grammatik, aus der spezifische XML-Vokabulare abgeleitet werden können
 - universelle “Metasprache“
 - Auszeichnungssprache:
 - textbasiertes Format
 - Informationen über Inhalte

XML-Dokument: Beispiel

```
<?xml version="1.0" encoding="UTF-8" ?>
<städte>
  <stadt xml:id="C" url="http://www.chemnitz.de">
    <name>Chemnitz</name>
    <bundesland>Sachsen</bundesland>
    <einwohner>240.000</einwohner>
    <fläche einheit="km²">220,85</fläche>
    <beschreibung>Chemnitz (1953–1990: Karl-Marx-Stadt) ist eine Großstadt
im Südwesten des Freistaates Sachsen. Mit mehr als 240.000 Einwohnern ist sie
nach <stadt ref="D">Dresden</stadt> und <stadt ref="L">Leipzig</stadt> die
drittgrößte Stadt und eines der sechs Oberzentren des Freistaates.
    </beschreibung>
  </stadt>
<!-- weitere Städte ergänzen! -->
</städte>
```

XML-Dokument: Elemente



```
<stadt>
  <name>Chemnitz</name>
</stadt>
```

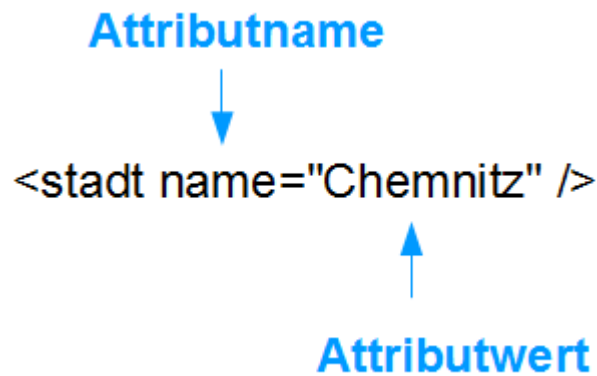
- Tag
 - = Element
 - Aufbau:
 - Start-Tag + Inhalt + End-Tag
 - Syntax:
 - `<name>...</name>`
 - `<name />`
 - Taginhalt:
 - Text
 - Elemente
 - beides („mixed content“)
 - nichts (leeres Element)

XML-Dokument: Elemente

- Regeln für Elementnamen
 - ein Name muss mit einem Buchstaben, Unterstrich oder Doppelpunkt beginnen
 - danach auch Zahlen, Bindestrich, Punkt erlaubt
 - „xml“ nie am Anfang
 - Länge beliebig
 - Groß- / Kleinschreibung ist relevant

- Semantische Tags?
 - <tag1>, <xyz> geht auch
 - Ziel aber bedeutungsvolle Namen

XML-Dokument: Attribute



- **Attribut**
 - = Element
 - **Aufbau:**
 - Attributname + Attributwert
 - **Syntax:**
 - name="wert"

XML-Dokument: Attribute

- Regeln für Attribute
 - Vergabe von Namen wie bei Elementen
 - ein Element
 - kann mehrere, verschiedene Attribute tragen
 - darf jedes Attribut nur einmal tragen
 - Attribute haben nur Textinhalt

- Attribute oder Unterelemente?
 - `<stadt url="http://www.chemnitz.de">...`
 - `<stadt>`
 - `<url>http://www.chemnitz.de</url>`
 - `</stadt>`



XML-Dokument: Attribute

- reservierte Attribute
 - xml:lang
 - „de“, „en“, ...
 - xml:space
 - „preserve“
 - „default“

XML-Dokument

XML-Deklaration



```
<?xml version="1.0" encoding="UTF-8" ?>
```

Kommentar

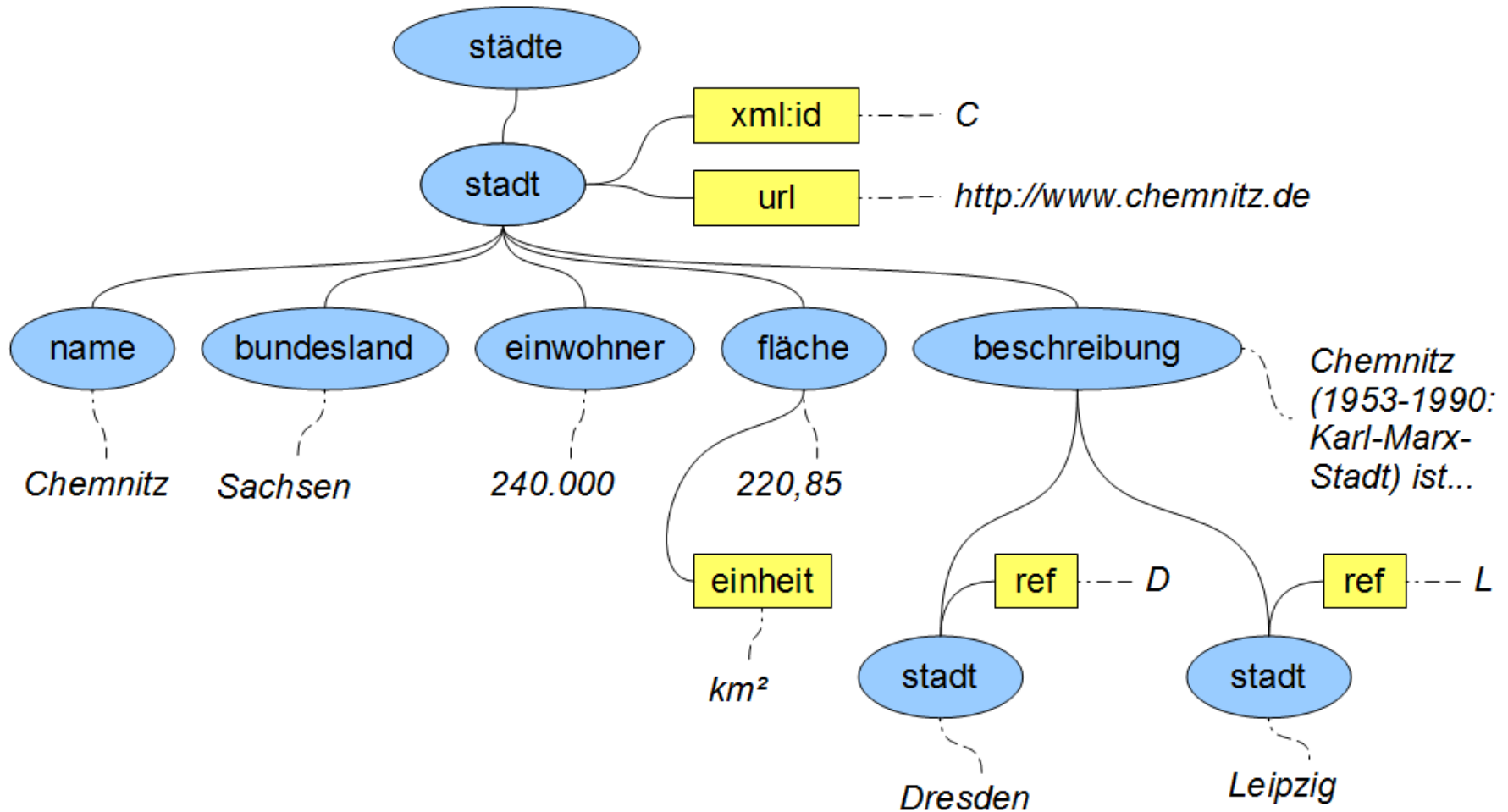


```
<!-- weitere Städte ergänzen! -->
```

- XML-Deklaration
 - = XML-Tag
 - „Processing Instruction“
 - Steht immer am Anfang eines XML-Dokuments

- Kommentar
 - Wird bei der Verarbeitung des Dokumentes ignoriert
 - Syntax:
 - <!-- *Kommentar* -->

XML-Baum





XML-Baum: Achsen

Achse	Name	Beschreibung
selbst	<i>self</i>	Der aktuelle Knoten (Kontextknoten)
Nachfahre	<i>descendant</i>	Alle Nachkommen (Elemente) des aktuellen Knotens
Vorfahre	<i>ancestor</i>	Alle Vorfahren (Elemente) des aktuellen Knotens bis zum Wurzelknoten
Kind	<i>child</i>	Kinder des aktuellen Knotens
Elternelement	<i>parent</i>	Eltern des aktuellen Knotens
folgende Elemente	<i>following</i>	Alle Elemente, die im Dokument auf den aktuellen Knoten folgen
vorausgehende Elemente	<i>preceding</i>	Alle Elemente, die dem aktuellen Knoten im Dokument vorausgehen
folgende Geschwister	<i>following-sibling</i>	Alle auf den aktuellen Knoten folgenden Geschwisterknoten.
vorausgehende Geschwister	<i>preceding-sibling</i>	Alle dem aktuellen Knoten vorausgehenden Geschwisterknoten
Attribut	<i>attribute</i>	Alle Attributknoten des aktuellen Elementknotens

XML-Baum: Pfade (Beispiele)

- / = einen Schritt weiter im Baum (nächster Lokalisierungsschritt)
- ::* = alle Elementtypen (kein bestimmter Elementname)
- @ = Attribut
- [Zahl] – das wievielte Element (aus der Ergebnismenge)
- text() = Textinhalt eines Elementes
- ::Elementtyp = Bedingung für den Elementtypen (Knotentest)
- // = beliebig tief im Baum
- [text() = ‚Textinhalt‘] = Bedingung (Prädikat)
- . = self

Gesucht	Pfad	Ergebnismenge
Alle Kinder von <städte>	/städte/child::*	<stadt>...</stadt>
Folgende Geschwister von <bundesland>	/städte/child::stadt/child::bundesland/ following-sibling::*	<einwohner>..., <fläche>..., <beschreibung>...
Alle vorausgehenden Elemente von <bundesland>	/städte/stadt/bundesland/preceding::*	<name>...</name>
Die Internetadresse	/städte/stadt/@url	„http://www.chemnitz.de“



XML-Baum: Pfade (Beispiele)

Gesucht	Pfad	Ergebnismenge
der Textknoten „Leipzig“	<code>/städte/stadt/beschreibung/stadt[2]/text()</code>	„Leipzig“
Von ‚Leipzig‘ über ‚Dresden‘ nach ‚Chemnitz‘	<code>/städte/stadt/beschreibung/stadt[2]/parent::beschreibung/stadt[1]/parent::beschreibung/preceding-sibling::name</code>	„Chemnitz“
Von ‚Leipzig‘ über ‚Dresden‘ nach ‚Chemnitz‘	<code>//stadt[2]/preceding::stadt[1]/preceding::name</code>	„Chemnitz“
Von ‚Leipzig‘ über ‚Dresden‘ nach ‚Chemnitz‘	<code>//stadt[text() = ‚Leipzig‘]/preceding::stadt[text() = ‚Dresden‘]/preceding::name[text() = ‚Chemnitz‘]</code>	„Chemnitz“
Von ‚Leipzig‘ über ‚Dresden‘ nach ‚Chemnitz‘	<code>//stadt[. = ‚Leipzig‘]/preceding::stadt[. = ‚Dresden‘]/preceding::name[. = ‚Chemnitz‘]</code>	„ http://www.chemnitz.de “
Alle Textknoten, die Nachfahren von <beschreibung> sind.	<code>//beschreibung/descendant::text()</code>	5 Textknoten!

Wohlgeformtes XML

- Wohlgeformtheit
 - mindestens ein Element auf höchster Ebene („Wurzelelement“), aber nur eines!
 - alle Tags müssen geschlossen sein
 - die Tags müssen richtig geschachtelt sein
 - Das geht: `<fläche>220,85 <einheit>km2</einheit></fläche>`
 - Das nicht: `<fläche>220,85 <einheit></fläche>km2</einheit>`
 - Groß- und Kleinschreibung werden unterschieden!
 - `<fläche>` vs. `<Fläche>`
 - die Regeln für Element- und Attributnamen sind zu beachten
 - Aber:
 - Besser `<staedte>` als `<städte>`
 - Besser `<email>` als `<e-mail>`

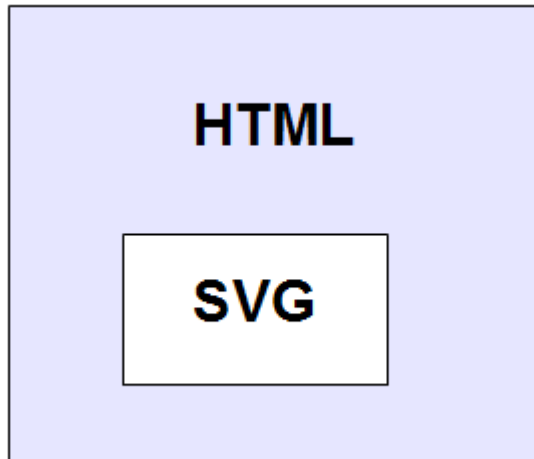
Wohlgeformtes XML

- ...
 - Sonderzeichen müssen als Entitätsreferenzen codiert werden, wenn sie Textinhalt sein sollen
 - < wird zu <
 - > wird zu >
 - ' wird zu '
 - " wird zu "
 - & wird zu &
 - Alternativ können sie als CDATA-Sektionen markiert werden
 - <![CDATA[Inhalte mit <spitzen> Klammern]]>

Gültiges XML

- Gültigkeit
 - = Validität
 - das XML-Dokument folgt einem Schema
 - es folgt der vorgegebenen Struktur
 - es enthält nur die in der Struktur erlaubten Tags und Attribute

Namensräume



Adressbuch

```
<adresse>  
  <name>...  
  <strasse>...  
  <nummer>...
```

Telefonbuch

```
<eintrag>  
  <name>...  
  <vorwahl>...  
  <nummer>...
```

Namensräume

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de">
  <head>
    <title>HTML-Datei mit Grafik</title>
  </head>
  <body>
    <svg:svg xmlns:svg="http://www.w3.org/2000/svg" version="1.1"
      width="700px" height="400px">
      <svg:line x1="0" y1="200" x2="700" y2="200" stroke="black" />
    </svg:svg>
  </body>
</html>
```

Namensräume

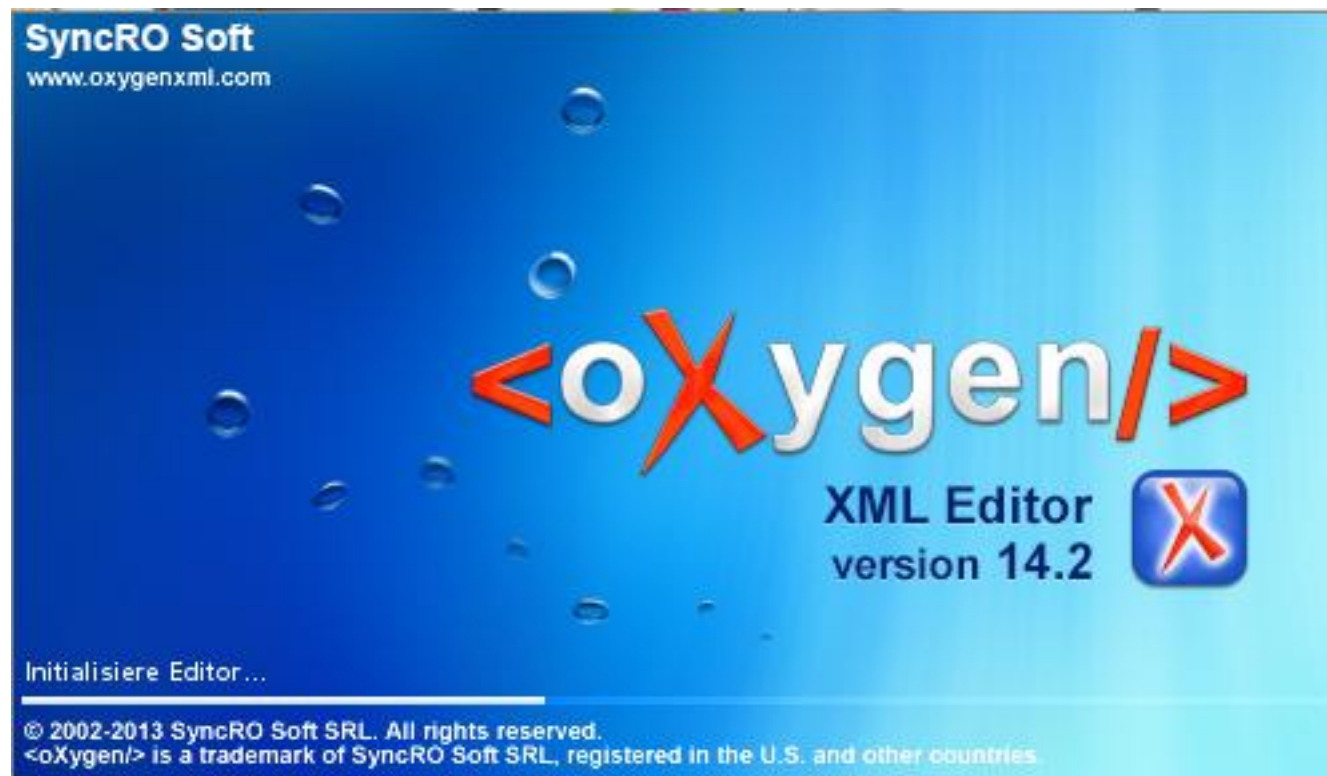
- Element- und Attributnamen sind Teil eines Namensraumes
- Namensraum
 - Sammlung von Namen zu einem Gegenstandsbereich
 - Identifikation über URI
- qualifizierter Elementname
 - `<präfix:name>`
 - Z.B. `<tei:p>`
- Deklaration
 - `<element xmlns="URI">...`
 - `<präfix:element xmlns:präfix="URI">...`
 - Z.B. `<tei:p xmlns:tei="http://www.tei-c.org/ns/1.0">...`

XML-Bearbeitung

- einfache Textprogramme
- XML-Editoren
 - erleichtern die Arbeit mit XML
 - Z.B.
 - XML Notepad
 - <http://www.microsoft.com/en-us/download/details.aspx?id=7973>
 - XMLSpy
 - <http://www.altova.com/de/xmlspy.html>
 - oXygen
 - <http://oxygenxml.com/>

oXygen

- Download XML-Editor: http://oxygenxml.com/download_oxygenxml_editor.html
- Testlizenz: <http://oxygenxml.com/register.html>



XML – Übungen (1)

1. XML-Dokumente
 1. Öffnen Sie die Datei „staedte.xml“ in oXygen und ergänzen Sie sie um zwei weitere Städte.
 2. Fügen Sie in „staedte.xml“ neue Element- und /oder Attributtypen mit weiteren Inhalten hinzu.

XML – Übungen (2)

1. Pfade

1. Vollziehen Sie die Pfadbeispiele von Folie 13 und 14 in oXygen nach.
2. Öffnen Sie die Datei „staedte-erweitert.xml“ und suchen Sie folgende Ergebnismengen mit Pfadausdrücken (Wie lauten die Pfadausdrücke?):
 1. Alle <stadt>-Elemente, die im Dokument vorkommen. Wie viele sind es?
 2. Lassen Sie sich den Namen der Stadt ausgeben, deren Fläche 105,40 km² beträgt.
 3. Lassen Sie sich den Textknoten der fünften Partnerstadt von London ausgeben.
 4. Geben sie die id (-> Attribut) von Paris aus.

XML – Übungen (3)

1. Wohlgeformtheit
 1. Suchen Sie alle Fehler in der Datei „o-me-o-life.xml“. Wie viele und welche sind es?
 2. Korrigieren Sie die Fehler.
2. Namensräume
 1. Öffnen Sie die Datei „artikel.xml“. Die bild-Elemente (mit Unterelementen) sollen einem anderen Namensraum zugeordnet werden als die übrigen, in <text> enthaltenen Elemente. Auch die Metadaten sollen einen eigenen Namensraum bekommen. Deklarieren Sie folgende Namensräume (mit Präfix) und weisen Sie sie den entsprechenden Elementen zu:
 1. <http://www.artikel.de/ns/metadaten>
 2. <http://www.artikel.de/ns/bilder>

XML – Literatur

- Vonhoegen, Helmut, *Einstieg in XML. Grundlagen, Praxis, Referenz*, Galileo Press, Bonn 2009⁵.
- St. Laurent, Simon/Fitzgerald, Michael, *XML. kurz & gut*, O'Reilly, Köln 2006³.
- Kay, Michael, *XSLT 2.0 and XPath 2.0 Programmer's Reference*, Wrox 2008⁴.
- W3 Schools Tutorien: <http://www.w3schools.com/xml/> (XML), <http://www.w3schools.com/xpath/> (XPath), <http://www.w3schools.com/html/> (HTML), <http://www.w3schools.com/css/> (CSS)
- W3C Spezifikation: <http://www.w3.org/TR/xml/>