

# Kapitel 10: XML

# XML

- 1969 GML Generalized Markup Language  
(Goldfarb/Mosher/Lorie)
- 1986 SGML Standardized Markup Language
- 1989 WWW World Wide Web
- 1992 HTML Hypertext Markup Language
- 1998 XML eXtensible Markup Language

# XML

```
<Titel>Selber Atmen</Titel>  
<Autor>Willi Wacker</Autor>
```

# XML: Strukturierte Daten

```
<Bücher>  
  <Titel>Selber Atmen</Titel>  
  <Autor>Willi Wacker</Autor>  
  <Titel>Richtig Streiken</Titel>  
  <Autor>Erika Mustermann</Autor>  
  <Titel>Schneller Beten</Titel>  
  <Autor>Daniel Düsentrieb</Autor>  
</Bücher>
```

Titel	Autor
Selber Atmen	Willi Wacker
Richtig Streiken	Erika Mustermann
Schneller Beten	Daniel Düsentrieb

# XML: Strukturierte Texte

## Selber Atmen

Willi Wacker

### 1. Einleitung

In einer Reihe von aufsehenerregenden Experimenten wurden kürzlich am *Max-Planck-Institut* die Vorteile des selbständigen Denkens herausgearbeitet.

Unsere Forschungen lassen erwarten, daß analoge Aussagen auch für den Atmungsvorgang gelten könnten.

```
<artikel>
  <Kopf>
    <Titel>Selber Atmen</Titel>
    <Autor>Willi Wacker</Autor>
  </Kopf>
  <Rumpf>
    <Kapitel>
      <Titel>Einleitung</Titel>
      <Absatz>
        In einer Reihe von aufsehenerregenden Experimenten
        wurden kürzlich am <betont>Max-Planck-Institut</betont>
        die Vorteile des selbständigen Denkens herausgearbeitet.
      </Absatz>
      <Absatz>
        Unsere Forschungen lassen erwarten, daß analoge
        Aussagen auch für den Atmungsvorgang gelten könnten.
      </Absatz>
    </Kapitel>
  </Rumpf>
</artikel>
```

# Dateien

<code>artikel.dtd</code>	Document Type Definition mit Strukturbeschreibung
<code>artikel.xml:</code>	XML-Datei mit den durch Tags markierten Daten.
<code>artikel.xsl:</code>	Stylesheet mit Angaben zum Rendern des Layout

# artikel.dtd

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!ELEMENT artikel      (Kopf,Rumpf)>
<!ELEMENT Kopf        (Titel,Autor)>
<!ELEMENT Rumpf       (Kapitel*)>
<!ELEMENT Kapitel     (Titel, Absatz*)>
<!ELEMENT Titel       (#PCDATA)>
<!ELEMENT Autor       (#PCDATA)>
<!ELEMENT Absatz     (#PCDATA|betont)*>
<!ELEMENT betont     (#PCDATA)>
```

# artikel.xml

```
<?xml version='1.0' encoding="ISO-8859-1" ?>
<!DOCTYPE artikel SYSTEM "artikel.dtd" >
<artikel>
  <Kopf>
    <Titel>Selber Atmen</Titel>
    <Autor>Willi Wacker</Autor>
  </Kopf>
  <Rumpf>
    <Kapitel>
      <Titel>Einleitung</Titel>
      <Absatz>
        In einer Reihe von aufsehenerregenden Experimenten wurden kürzlich am
        <betont>Max-Planck-Institut</betont> die Vorteile des selbständigen Denkens
        herausgearbeitet.
      </Absatz>
      <Absatz>
        Unsere Forschungen lassen erwarten, daß analoge Aussagen auch für den Atmungsvorgang
        gelten könnten.
      </Absatz>
    </Kapitel>
    <Kapitel>
      <Titel>Ausblick</Titel>
      <Absatz>
        Es gibt viel zu tun; warten wir es ab !
      </Absatz>
    </Kapitel>
  </Rumpf>
</artikel>
```

# artikel.xsl (Teil 1)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="artikel">
    <html>
      <head><title><xsl:value-of select="Titel"/></title></head>
      <body><xsl:apply-templates/></body>
    </html>
  </xsl:template>

  <xsl:template match="artikel/Kopf/Titel">
    <center><h1><xsl:apply-templates/></h1></center>
  </xsl:template>

  <xsl:template match="artikel/Kopf/Autor">
    <center><h4><xsl:apply-templates/></h4></center>
  </xsl:template>
```

# artikel.xsl (Teil 2)

```
<xsl:template match="artikel/Rumpf/Kapitel/Titel">
  <h3><xsl:apply-templates/></h3>
</xsl:template>
```

```
<xsl:template match="Absatz">
  <p><xsl:apply-templates/></p>
</xsl:template>
```

```
<xsl:template match="betont">
  <I><xsl:apply-templates/></I>
</xsl:template>
```

```
</xsl:stylesheet>
```

[Validator](#)

[Transformer](#)

# personen.dtd

```
<?xml version='1.0' encoding="ISO-8859-1" ?>
<!ELEMENT personen          (person*)>
<!ELEMENT person           (vorname,nachname,photo)>
<!ATTLIST person           persnr      ID #REQUIRED>
<!ATTLIST person           chef        IDREF #IMPLIED>
<!ATTLIST person           geschlecht  (männlich|weiblich) #REQUIRED>
<!ELEMENT vorname          (#PCDATA)>
<!ELEMENT nachname         (#PCDATA)>
<!ELEMENT photo            EMPTY>
<!ATTLIST photo            src         CDATA #REQUIRED>
```

# personen.xml

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE personen SYSTEM "personen.dtd">
<personen>
  <person persnr="P4711" chef="P4712" geschlecht="weiblich" >
    <vorname>Erika</vorname>
    <nachname>Mustermann</nachname>
    <photo src="erika.gif"/>
  </person>
  <person persnr="P4712" geschlecht="maennlich">
    <vorname>Willi</vorname>
    <nachname>Wacker</nachname>
    <photo src="willi.gif"/>
  </person>
</personen>
```

# dozenten.dtd

```
<?xml version='1.0' encoding="ISO-8859-1" ?>
<!ELEMENT dozenten      (dozent*)>
<!ELEMENT dozent        (PersNr,Name,Rang,Raum,Amt*)>
<!ELEMENT PersNr        (#PCDATA)>
<!ELEMENT Name          (#PCDATA)>
<!ELEMENT Rang          (#PCDATA)>
<!ELEMENT Raum          (#PCDATA)>
<!ELEMENT Amt           (Bezeichnung, Termin?)>
<!ELEMENT Bezeichnung   (#PCDATA)>
<!ELEMENT Termin       (#PCDATA)>
```

Beschreibung durch Schema: [dozenten.xsl](#)

# dozenten.xml (Teil 1)

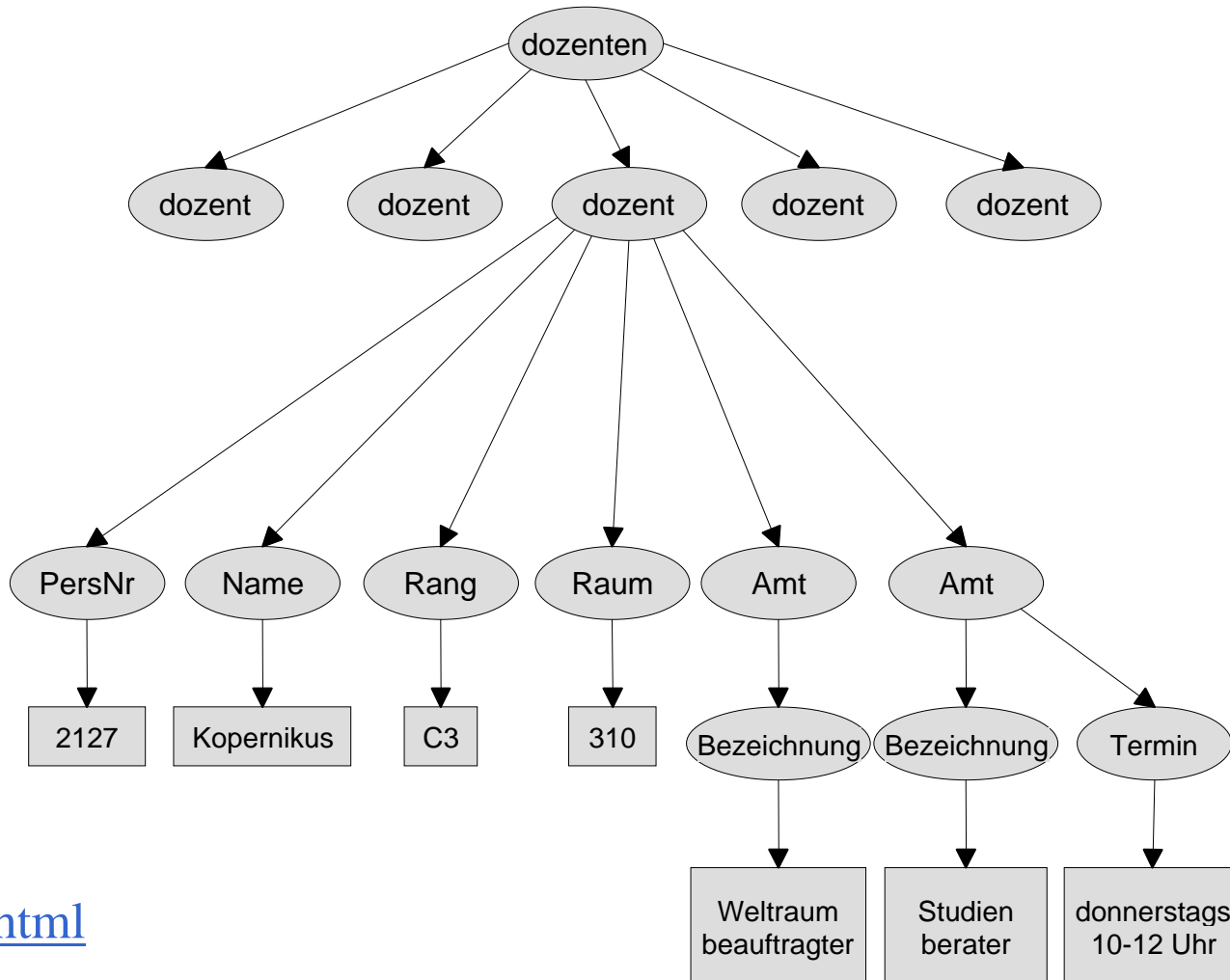
```
<?xml version='1.0' encoding="ISO-8859-1" ?>
<!DOCTYPE dozenten SYSTEM "dozenten.dtd" >
<dozenten>
  <dozent>
    <PersNr>2125</PersNr>
    <Name>Sokrates</Name>
    <Rang>C4</Rang>
    <Raum>226</Raum>
    <Amt>
      <Bezeichnung>Dekan</Bezeichnung>
      <Termin>vormittags</Termin>
    </Amt>
  </dozent>
  <dozent>
    <PersNr>2126</PersNr>
    <Name>Russel</Name>
    <Rang>C4</Rang>
    <Raum>232</Raum>
  </dozent>
</dozenten>
```

# dozenten.xml (Teil2)

```
<dozent>
  <PersNr>2127</PersNr>
  <Name>Kopernikus</Name>
  <Rang>C3</Rang>
  <Raum>310</Raum>
  <Amt>
    <Bezeichnung>Weltraumbeauftragter</Bezeichnung>
  </Amt>
  <Amt>
    <Bezeichnung>Studienberater</Bezeichnung>
    <Termin>donnerstags 10-12 Uhr</Termin>
  </Amt>
</dozent>
...

</dozenten>
```

# XML-Baum



# XML vom SQL Server 2000

```
select name, persnr  
from professoren  
for xml auto, elements
```

Query-Analyzer

# XQuery

- FLWR-Ausdruck (FOR , LET , WHERE , RETURN)
- Pfadausdruck in XPath
- Element-Konstrukturen

# XQuery (FLWR)

Liste alle Dozenten, die mindestens eine Vorlesung halten:

```
<aktive-Dozenten>
{FOR $d IN document("dozenten.xml")//dozent
 LET $v := document("vorlesungen.xml")
   //vorlesung[gelesenVon=$d/PersNr]
 WHERE count($v) > 0
 RETURN <Dozent>$d/Name</Dozent> }
</aktive-Dozenten>
```

```
<aktive-Dozenten>
  <Dozent>
    <name>
      Sokrates
    </name>
  </Dozent>
  <Dozent>
    <name>
      Kant
    </name>
  </Dozent>
  <Dozent>
    <name>
      Russel
    </name>
  </Dozent>
</aktive-Dozenten>
```

# Xquery (Join)

Liste alle Dozenten mit ihren Lehrveranstaltungen:

```
<Vorlesungsverzeichnis>
  {FOR $d IN document("dozenten.xml")//dozent,
    $v IN document("vorlesungen.xml")//vorlesung
  WHERE $d/PersNr = $v/gelesenVon
  RETURN <Veranstaltung>$d/Name, $v/Titel</Veranstaltung>}
</Vorlesungsverzeichnis>
```

# XQuery

Liste zu jedem Dozenten seine Lehrveranstaltungen:

```
<Vorlesungsverzeichnis>
  {FOR $d IN document("dozenten.xml")//dozent
  RETURN
    <Dozent>$d/Name</Dozent>
    {FOR $v IN document("vorlesungen.xml")//vorlesung
    WHERE $d/PersNr = $v/gelesenVon
    RETURN <Vorlesung>$v/Titel</Vorlesung>}}
</Vorlesungsverzeichnis>
```

# XQuery (Aggregate Function)

Liste zu jedem Dozenten die Summe der Semesterwochenstunden seiner Lehrveranstaltungen:

```
<Dozenten>
```

```
{FOR $d IN document("dozenten.xml")//dozent
  LET $s := sum(document("vorlesungen.xml")
                //vorlesung[gelesenVon=$d/PersNr]/SWS)
  WHERE $s > 0
  RETURN <Dozent>
         <Name>$d/Name</Name>
         <Lehrbelastung>$s</Lehrbelastung>
       </Dozent>}
```

```
</Dozenten>
```

# XQuery (Self Join)

Liste alle Dozenten mit demselben Rang wie Sokrates:

```
<Rang-wie-Sokrates>  
  {FOR $d1 IN document("dozenten.xml")//dozent[Name="Sokrates"],  
    $d2 IN document("dozenten.xml")//dozent[Rang=$d1/Rang]  
    RETURN <Name>$d2/Name</Name>}  
</Rang-wie-Sokrates>
```

# Apache Xindice

<http://xml.apache.org/xindice>

*"If you don't have any XML data, don't want any XML data or think XML is the most over-hyped technology of the new millennium, then Xindice is not for you"*

# Web Services

Bisher: Google +  
von Menschen besuchte Webseiten

Jetzt: Verzeichnisdienst +  
maschinenlesbare Dokumente

UDDI: Universal Description, Discovery and Integration

WSDL: Web Services Description Language

SOAP: Simple Object Access Protokoll

# Amazon Web Service

WSDL-Datei:

<http://soap.amazon.com/schemas2/AmazonWebServices.wsdl>

Suche nach Harry Potter:

<http://webservices.amazon.de/onca/xml?Service=AWSECommerceService&SubscriptionId=0BB9F8ZNDA7HG04QG1G2&Operation=ItemSearch&SearchIndex=Books&Keywords=Harry%20Potter&ResponseGroup=Small>

# UniVerwaltung Web Service

WSDL-Datei:

<http://www-lehre.inf.uos.de/~dbs/2005/WS/UniVerwaltung/UniVerwaltung.wsdl>

Website zum Testen von Web Services:

<http://www.mindreef.net>

# Request

```
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getLehrUmfangVonProfessor
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="UniVerwaltung">
      <ProfName xsi:type="xsd:string">Sokrates</ProfName>
    </ns1:getLehrUmfangVonProfessor>
  </soapenv:Body>
</soapenv:Envelope>
```

# Response

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope
  xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getLehrUmfangVonProfessorResponse
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:ns1="UniVerwaltung">
      <LehrUmfang href="#id0"/>
    </ns1:getLehrUmfangVonProfessorResponse>
    <multiRef id="id0"
      soapenc:root="0"
      soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xsi:type="xsd:int"
      xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
      10
    </multiRef>
  </soapenv:Body>
</soapenv:Envelope>
```

# InquireDB.java

```
public static int getLehrUmfangVonProfessor(String profName) {
    int sumsws = 0;
    String url = "jdbc:microsoft:sqlserver://maestro:1433";
    String user = "erika"; String pwd = "mustermann";
    try { Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver");
        Connection con = DriverManager.getConnection(url, user, pwd);
        PreparedStatement stmt = con.prepareStatement(
            "SELECT SUM(v.SWS) AS SUMSWS FROM Vorlesungen v,Professoren p " +
            "WHERE v.gelesenVon = p.PersNr " + "AND p.Name = ?");
        stmt.setString(1, profName);
        ResultSet rset = stmt.executeQuery();
        rset.next();
        sumsws = java.lang.Integer.parseInt(rset.getString("SUMSWS"));
        rset.close();
        stmt.close();
        con.close();
    } catch (Exception e) {e.printStackTrace();}
    return sumsws;
}
```

SSH: [ClientUniverwaltung.java](#)