

Vorlesungsplan

1. Übersicht
2. Datenmodelle
 - Textdateien, Entry-Modell
 - Relationale DB
 - XML
3. Genom-DB
4. Genexpressions-DB
5. Protein-DB
6. Pathway-DB
7. Publikations-DB
8. Zugriff und Anfragesysteme
9. Datenintegration von Bio-DB
10. Webservices

XML Überblick

- XML Syntax
- XML Namespaces
- DTD
- XPath

Syntax eines XML Dokuments

- XML Dokumente sind selbstbeschreibend

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```

- Zeile 1: XML Deklaration, Version, Char-Coding

XML Validierung

- "Well Formed" XML Dokumente

- Selbstbeschreibend
- Klammern
- Test: http://www.w3schools.com/dom/dom_validate.asp

- Gültige XML Doks

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE note SYSTEM "InternalNote.dtd">
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```
- entsprechen einer DTD
- DTD definiert Bausteine
- XML Schema
 - Alternative zu DTD

XML bearbeiten

- Ansehen
 - Browser: Firefox, Mozilla, Opera 7&8, IE
- XSL kann XML Dokumente transformieren
 - auch XML in HTML
- Editoren
 - jeder Texteditor funktioniert
 - XML Editoren
 - automatisch schliessende Tags
 - erzwingt gültiges XML
 - Überprüft XML anhand von DTD oder XML Schema
 - Altova's XMLSpy
 - Open XML Editor <http://www.philo.de/xmledit/>
 - ...

XML Namespaces

- Namenskonflikte

<pre><table> <tr> <td>Apples</td> <td>Bananas</td> </tr> </table></pre>	<pre><table> <name>African Coffee Table</name> <width>80</width> <length>120</length> </table></pre>
---	--
- Auflösung von Konflikten durch Präfix

<pre><h:table xmlns:h="http://www.w3.org/TR/html4/"> <h:tr> <h:td>Apples</h:td> <h:td>Bananas</h:td> </h:tr> </h:table></pre>	<pre><f:table xmlns:f="http://www.w3schools.com/furniture"> <f:name>African Coffee Table</f:name> <f:width>80</f:width> <f:length>120</f:length> </f:table></pre>
---	---
- XML Namespace (xmlns) Attribut
 - xmlns:namespace-prefix="namespaceURI",
 - Default NS: xmlns="namespaceURI"

<pre><table xmlns="http://www.w3.org/TR/html4/"> <tr> <td>Apples</td> <td>Bananas</td> </tr> </table></pre>	<pre><table xmlns="http://www.w3schools.com/furniture"> <name>African Coffee Table</name> <width>80</width> <length>120</length> </table></pre>
---	---

Verwendung von NS

- Häufig in XSL-Dok. zu finden

```
<?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="/">

<html> <body>
<h2>My CD Collection</h2>
<table border="1">
<tr> <th align="left">Title</th> <th align="left">Artist</th> </tr>

<xsl:for-each select="catalog/cd">
<tr> <td><xsl:value-of select="title"/></td> <td><xsl:value-of select="artist"/></td> </tr>
</xsl:for-each>

</table>
</body> </html>

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

XML und Sonderzeichen

- Sonderzeichen
 - < less than
 - > greater than
 - & ampersand
 - ' apostrophe
 - " quotation mark
- CDATA
 - <![CDATA[alles innerhalb wird ignoriert]]>

```
<script>
<![CDATA[
function matchwo(a,b)
{
if (a < b && a < 0) then { return 1 }
else { return 0 } }
]]>
</script>
```

DTD

- DTD

- Document Type Definition definiert die Bausteine eines XML Dokuments.
- Es definiert die Struktur mit einer Liste erlaubter Elemente.
- `<!DOCTYPE root-element [element-declarations]>`

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <ELEMENT note (to,from,heading,body)>
  <ELEMENT to (#PCDATA)>
  <ELEMENT from (#PCDATA)>
  <ELEMENT heading (#PCDATA)>
  <ELEMENT body (#PCDATA)> ]>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend</body>
</note>
```

- Externe Definition: `<!DOCTYPE root-element SYSTEM "filename">`
- Types: PCDATA (Parsed Data), CDATA (Char-Data)

DTD: Daten-Elemente

- Declaration

- `<!ELEMENT element-name category>`
 - Kategorie: EMPTY oder ANY
- `<!ELEMENT element-name (element-content)>`
- `<!ELEMENT element-name (child-element-name,child-element-name,.....)>`
 - Reihenfolge muß im XML Dok. beachtet werden

- Deklaration, daß ein Element

- genau einmal auftritt
 - `<!ELEMENT element-name (child-name)>`
- mindestens ein oder mehrmals auftritt
 - `<!ELEMENT element-name (child-name+)>`
- kein oder mehrmals auftritt
 - `<!ELEMENT element-name (child-name*)>`
- ein oder keinmal auftritt
 - `<!ELEMENT element-name (child-name?)>`
- entweder oder auftritt
 - `<!ELEMENT note (to,from,header,(message|body))>`
- Gemischt
 - `<!ELEMENT note (#PCDATA|to|from|header|message)*>`

DTD: Attribute

- Deklaration

- Allgemein: `<!ATTLIST element-name attribute-name attribute-type default-value>`
- Beispiel: `<!ATTLIST payment type CDATA "check">`
- In XML `<payment type="check" />`

Attribut-Typen und Default-Werte

- Attribut-Typen

- | | |
|--------------------|---|
| – CDATA | The value is character |
| – data(en1 en2 ..) | The value must be one from an enumerated list |
| – ID | The value is a unique id |
| – IDREF | The value is the id of another element |
| – IDREFS | The value is a list of other ids |
| – NMTOKEN | The value is a valid XML name |
| – NMTOKENS | The value is a list of valid XML names |
| – ENTITY | The value is an entity |
| – ENTITIES | The value is a list of entities |
| – NOTATION | The value is a name of a notation |
| – xml: | The value is a predefined xml value |

- Default-Werte

- | | |
|----------------|---|
| – value | The default value of the attribute |
| – #REQUIRED | The attribute value must be included in the element |
| – #IMPLIED | The attribute does not have to be included |
| – #FIXED value | The attribute value is fixed |

Beispiele: http://www.w3schools.com/dtd/dtd_attributes.asp

DTD: Entities

- Syntax
 - <!ENTITY entity-name "entity-value">
- DTD Beispiele
 - <!ENTITY writer "Donald Duck.">
 - <!ENTITY copyright "Copyright W3Schools.">
- XML Beispiel
 - <author>&writer;©right;</author>
- Externe Deklaration
 - <!ENTITY entity-name SYSTEM "URI/URL">

XPath

- XPath ist eine Sprache zum Finden von Informationen in einem XML Dokument. XPath wird zum Navigieren durch Elemente und Attribute in einem XML Dokument genutzt.
- XPath ist ein Hauptelement im W3C's XSLT standard - außerdem XQuery und XPointer bauen auf XPath Ausdrücke auf.
- XPath
 - ist eine Syntax mit der Teile eines XML Dok. definiert werden können

XPath Knoten

- Es gibt 7 Arten von Knoten
 - Element, Attribut, Text, Namespace, Processing-Instruction, Kommentar und Dokument (root) Knoten
- Beispiel

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
<book>
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
</bookstore>
```
- Knoten

```
<bookstore> (Dokument Knoten)
<author>J K. Rowling</author> (Elementknoten)
lang="en" (Attributknoten)
```
- Atomare Werte

```
J K. Rowling
"en"
```

Beziehungen von Knoten

- Eltern

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
<book>
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
</bookstore>
```

 - Book ist Eltern-Teil von title, author, year und price
- Kinder
 - title, author, year und price sind Kinder von book
- Geschwister
 - title, author, year und price sind Geschwister
- Vorfahren
 - book und bookstore sind Vorfahren von title
- Nachfahren
 - Nachfahren von bookstore sind book, title, author, year und price

Beispiel

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book>
    <title lang="eng">Harry Potter</title>
    <price>29.99</price>
  </book>
  <book>
    <title lang="eng">Learning XML</title>
    <price>39.95</price>
  </book>
</bookstore>
```

XPath Syntax

- Ausdrücke
 - nodename Selektiert alle Kinder des Knotens
 - / Selektiert die Wurzel
 - // Selektiert Nachkommen
 - . Selektiert den akt. Knoten
 - .. Selektiert das Elternteil des akt. Knotens
 - @ Selektiert Attribute
- Beispiele
 - bookstore Selektiert alle Kinder von bookstore element
 - /bookstore Selektiert die Wurzel bookstore
Beachte: Falls der Pfad mit einem Slash (/) beginnt, ist es ein absoluter Pfad zu einem Element!
 - bookstore/book Selektiert alle book Elemente, die Kinder von bookstore sind
 - //book Selektiert alle book Elemente
 - bookstore//book Selektiert alle book Elements, die Nachkommen von bookstore sind
 - //@lang Selektiert alle Attribute mit dem Namen lang

XPath Syntax (2)

- Prädikate
 - /bookstore/book[1] Selektiert das erste book Element, das Kind des bookstore Elements ist
 - /bookstore/book[last()] Selektiert das letzte book Element, das Kind des bookstore Elements ist
 - /bookstore/book[last()-1] Selektiert das vorletzte book Element, das Kind des bookstore Elements ist
 - /bookstore/book[position()<3] Selektiert die ersten beiden book Elemente, die Kinder des bookstore Elements sind
 - //title[@lang] Selektiert alle title Elemente, die ein Attribut lang haben
 - //title[@lang='eng'] Selektiert alle title Elemente, die ein Attribut lang mit Wert 'eng' haben
 - /bookstore/book[price>35.00] Selektiert alle book Element des bookstore Elements, die ein price Element mit Wert > 35.00 haben
 - /bookstore/book[price>35.00]/title Selektiert alle title Elemente der book Elemente des bookstore Elements, die ein price Element mit Wert > 35.00 haben

XPath Syntax (3)

- Wildcards
 - * Matcht jeden Elementknoten
 - @* Matcht jeden Attributknoten
 - node() Matcht jeden Knoten beliebigen Typs
- Beispiele
 - /bookstore/* Selektiert alle Kinderknoten des bookstore Elements
 - //* Selektiert alle Elemente im Dokument
 - //title[@*] Selektiert alle title Element, die ein beliebiges Attribut haben
- Alternative Pfade
 - werden mit | Operator spezifiziert
 - //book/title | //book/price Selektiert alle title UND price Elemente aller book Elemente
 - //title | //price Selektiert alle title AND price Elemente im Dokument
 - /bookstore/book/title | //price Selektiert alle title Element des book Element unter dem bookstore Element AND alle price Elements im Dokument

XPath Achsen

- Eine Achse definiert Knotenmengen relativ zum akt. Knoten
 - ancestor Selects all ancestors (parent, grandparent, etc.) of the current node
 - ancestor-or-self Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself
 - attribute Selects all attributes of the current node
 - child Selects all children of the current node
 - descendant Selects all descendants (children, grandchildren, etc.) of the current node
 - descendant-or-self Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself
 - following Selects everything in the document after the closing tag of the current node
 - following-sibling Selects all siblings after the current node
 - namespace Selects all namespace nodes of the current node
 - parent Selects the parent of the current node
 - preceding Selects everything in the document that is before the start tag of the current node
 - preceding-sibling Selects all siblings before the current node
 - self Selects the current node

XPath Achsen (2)

- Pfad Ausdruck
 - Absolut: /step/step/...
 - Relativ: step/step/...
 - Schritt kann sein
 - Achse
 - Knotentest
 - ein oder mehrere Prädikate
 - Syntax für eine Schritt
 - axisname::nodetest[predicate]

XPath Achsen, Beispiele

- child::book Selects all book nodes that are children of the current node
- attribute::lang Selects the lang attribute of the current node
- child::* Selects all children of the current node
- attribute::* Selects all attributes of the current node
- child::text() Selects all text child nodes of the current node
- child::node() Selects all child nodes of the current node
- descendant::book Selects all book descendants of the current node
- ancestor::book Selects all book ancestors of the current node
- ancestor-or-self::book Selects all book ancestors of the current node - and the current as well if it is a book node
- child::*/*child::price Selects all price grandchildren of the current node

Weitere Beispiele: http://www.w3schools.com/xpath/xpath_examples.asp