

XML – Eine Einführung

XML-Dokumente Im Internet Darstellen

XML – Eine Einführung

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-
Stylesheets Anzeigen

- Um was geht es in diesem Kapitel?
 - XSL – Extensible Stylesheet Language
 - Gestattet
 - eine flexiblere und leistungsfähigere Darstellung von XML-Dokumenten als mit CSS
 - Daten aus einem XML-Dokument zu selektieren und zu filtern
 - XSL regelt den Datenzugriff und Transformation (XSLT)
 - XSL ist komplexer als CSS (und mächtiger)
 - Links
 - <http://www.w3.org/TR/2001/REC-xsl-20011015>
 - <http://www.w3.org/TR/WD-xsl>
 - <http://www.w3.org/TR/2002/WD-xslt20-20020816/>
 - <http://www.w3.org/TR/WD-xslt>

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen

- Damit wir XSL einsetzen können, müssen wir in der Lage sein
 - XSL-Stylesheets zu erstellen
 - XSL-Dokumente sind wohlgeformte XML-Dokumente
 - XSL-Stylesheets mit XML-Dokumenten verknüpfen können
 - Dateityps
 - Anstelle von XML verwenden wir den Dateityp XSL
 - Verknüpfung von XML und XSL
 - mittels Verarbeitungsanweisung
 - » `<?xml version=„1.0“?>`
 - `<?xml-stylesheet type=„text/xsl“ href=XSLDateiPfad>`
 - ...
 - XSLDateiPfad kann auch eine URL sein (<http://../meinXSL.xml>)*

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Ein Beispiel

– Beispiel eines XSL's für die Darstellung von XML

- XML

- `<?xml version="1.0"?>`

- `<!-- Dateiname: XSL Intro.xml -->`

- `<?xml-stylesheet type="text/xsl" href="XSL Intro.xsl"?>`

- `<BOOK>`

- `<TITLE>Moby-Dick</TITLE>`

- `<AUTHOR>`

- `<FIRSTNAME>Herman</FIRSTNAME>`

- `<LASTNAME>Melville</LASTNAME>`

- `</AUTHOR>`

- `<BINDING>Taschenbuch</BINDING>`

- `<PAGES>336</PAGES>`

- `<PRICE>EUR 12,75</PRICE>`

- `</BOOK>`

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Ein Beispiel

- Beispiel eines XSL's für die Darstellung von XML

- XSL

- `<?xml version="1.0"?>`

```
<!-- Dateiname: XSL Intro.xsl -->
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
  <xsl:template match="/">
```

```
    <H2>Buchbeschreibung</H2>
```

```
    <SPAN STYLE="font-style:italic">Autor: </SPAN>
```

```
    <xsl:value-of select="BOOK/AUTHOR"/><BR/>
```

```
    <SPAN STYLE="font-style:italic">Titel: </SPAN>
```

```
    <xsl:value-of select="BOOK/TITLE"/><BR/>
```

```
    <SPAN STYLE="font-style:italic">Preis: </SPAN>
```

```
    <xsl:value-of select="BOOK/PRICE"/><BR/>
```

```
    <SPAN STYLE="font-style:italic">Bindung: </SPAN>
```

```
    <xsl:value-of select="BOOK/BINDING"/><BR/>
```

```
    <SPAN STYLE="font-style:italic">Seitenzahl: </SPAN>
```

```
    <xsl:value-of select="BOOK/PAGES"/>
```

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

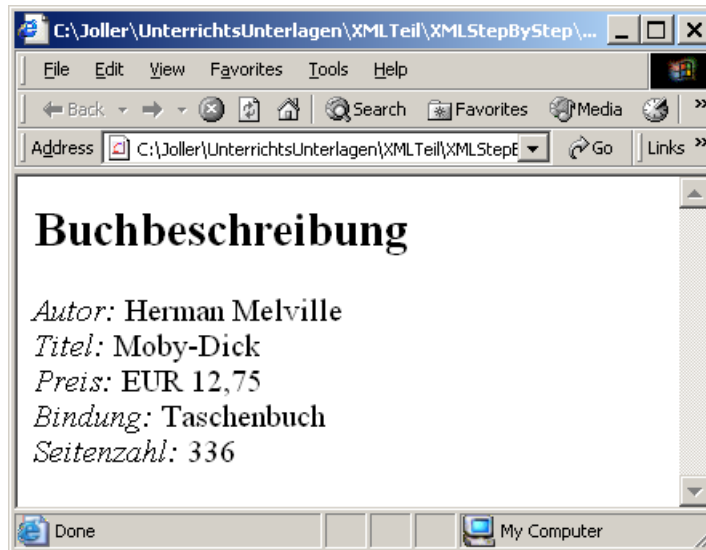
```
</xsl:stylesheet>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Ein Beispiel

- Beispiel eines XSL's für die Darstellung von XML
 - Ausgabe



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Ein Beispiel

- Erläuterungen zum XSL-Stylesheet
 - `<?xml version="1.0"?>`
 - XSL-Stylesheets sind XML-Dokumente
 - `<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">`
 - **XSL-Stylesheets identifizieren sich selbst als XSL-Stylesheets**
 - **Alle XSL-Elemente gehören zum Namensraum xsl, daher sind alle Namen der XSL-Elemente mit dem Präfix xsl versehen**
 - `<xsl:template match="/">...</xsl:template>`
 - Das XSL-Dokumentelement xsl-stylesheet muss eine oder mehrere XSL-Vorlagenelemente (Vorlagen) enthalten (in unserem Beispiel nur eine)
 - Der Browser verwendet Vorlagen, um Zweige des XML-Dokuments anzuzeigen oder auszublenden
 - Das Attribut match zeigt an, um welche Zweige es sich handelt
 - „/“ repäsentiert die Wurzel des gesamten XML-Dokuments (dies entspricht dem *Document*-Wurzelknoten im DOM)
 - In unserem Fall soll also das gesamte Dokument angezeigt werden

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Ein Beispiel

– Erläuterungen zum XSL-Stylesheet

- XSL-Vorlagen enthalten zwei Typen von XML-Elementen
 - **XML-Elemente, welche HTML-Tags repräsentieren**
 - » `<H2>Buchbeschreibung</H2>`
`Autor: `
`</BR>` `<!-- beachten Sie den Unterschied zum reinen HTML-Tag !- ->`
 - **XML-Elemente, welche XSL-Elemente darstellen**
 - » `<xsl:value-of select="BOOK/AUTHOR"/>`
- Der Browser erkennt die XSL-Elemente am Präfix *xsl*
 - *Der Browser erkennt, dass diese Elemente Anweisungen für die Bearbeitung des XML-Dokuments enthalten*
- Die HTML-Elemente werden (modifiziert) in die Ausgabe kopiert, wobei die Reihenfolge durch das XSL bestimmt wird, nicht wie bei CSS (Reihenfolge wie im XML)

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Ein Beispiel

– Erläuterungen zum XSL-Stylesheet

- **Das *value-of* XSL-Element**

- Fügt den Textinhalt des angegebenen XML-Elements (und *untergeordneter* Elemente ab dem im *template match* spezifizierten Element) zur HTML-Ausgabe hinzu
- Welche Elemente ausgewählt werden, steht im **select-Attribut** des **value-of –Elements**

- `<xsl:value-of select="BOOK/AUTHOR"/>`

- » Selektiere den Textinhalt des AUTHOR Elements im XML-Dokument und gib dessen Textinhalt aus
- » Falls das XML-Dokument folgenden Aufbau hat
`<AUTHOR>`
`<FIRSTNAME>Herman</FIRSTNAME>`
`<LASTNAME>Melville</LASTNAME>`

werden FIRSTNAME und LASTNAME ausgegeben

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Anzeige mehrerer Elemente : *for-each* XSL-Element

– Motivation

- Wenn Sie ein komplexeres XML-Dokument mit mehreren Bucheinträgen darstellen wollen, reicht das vorgestellte XSL-Stylesheet nicht
 - Es wird jeweils nur genau ein Buch dargestellt
- XSL kennt aber Kontrollstrukturen, analog zu Schleifen in normalen Programmiersprachen (while(..); for(...))

– XSL-Fragment

- ..

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<H2>Buchbestand</H2>
<xsl:for-each select="INVENTORY/BOOK">
  <SPAN STYLE="font-style:italic">Titel: </SPAN>
  <xsl:value-of select="TITLE"/><BR />
  ...
</xsl:for-each>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Anzeige mehrerer Elemente : *for-each* XSL-Element

– Das XSL-*for-each* Element bewirkt zweierlei

- Ausgabe des Elementblocks innerhalb des *for-each*-Elements für jedes XML-Element im XML-Dokument, das dem Muster entspricht (*select-Attribut* des *for-each-Elements*)
- Innerhalb des *for-each*-Elements entspricht das aktuelle Element demjenigen, das vom *select-Attribut* des *for-each-Elements* spezifiziert wird.
(im Beispiel: /INVENTORY/BOOK)
Die Ausgabe zeigt also alle Bücher an

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Anzeige mehrerer Elemente : *for-each* XSL-Element

- XML-Fragment

- ..

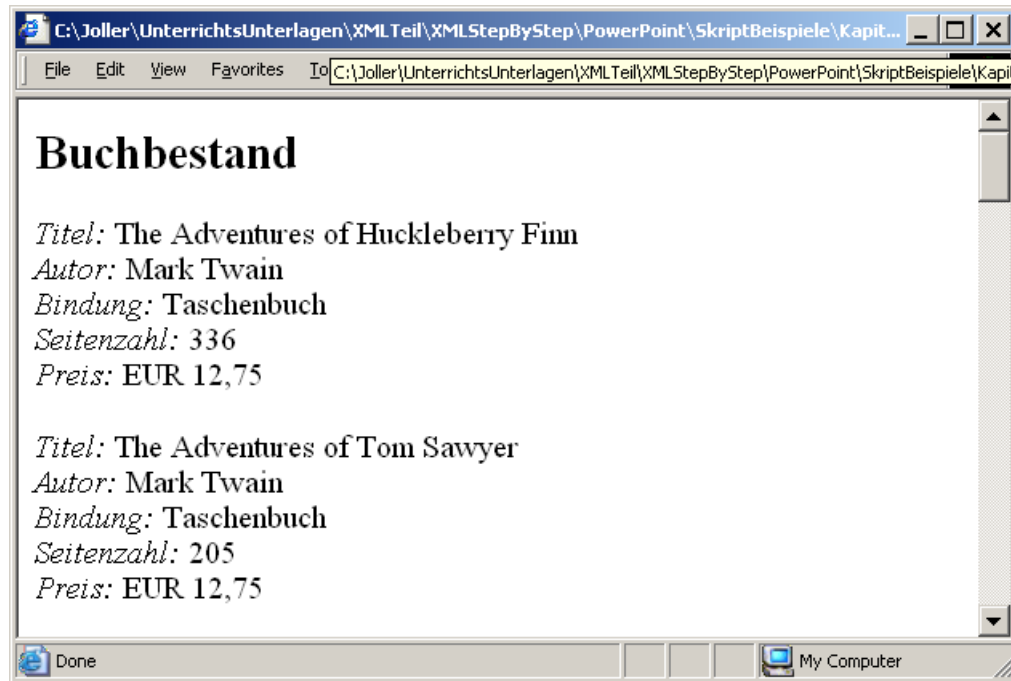
```
<?xml-stylesheet type="text/xsl" href="XSL for each.xsl"?>
<INVENTORY>
  <BOOK>
    <TITLE>The Adventures of Huckleberry Finn</TITLE>
    <AUTHOR>
      <FIRSTNAME>Mark</FIRSTNAME>
      <LASTNAME>Twain</LASTNAME>
    </AUTHOR>
    <BINDING>Taschenbuch</BINDING>
    <PAGES>336</PAGES>
    <PRICE>EUR 12,75</PRICE>
  </BOOK>
  <BOOK>
    <TITLE>The Adventures of Tom Sawyer</TITLE>
    ...
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Anzeige mehrerer Elemente : *for-each* XSL-Element

– Ausgabe



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Grundlagen – Mehrere Vorlagen (XSL *template*-Elemente)

– Motivation

- Die Kombination mehrere XSL Vorlagen kann sehr hilfreich sein, wenn Sie beispielsweise für eine Firma mit mehreren Produktbereichen übergeordnete und produktspezifische XSL's definieren müssen
- Durch die Kombination mehrerer XSL Vorlagen können interessante Effekte erzeugt werden
- XSL kennt aber auch Kontrollstrukturen, neben *for-each*, analog zu Schleifen in normalen Programmiersprachen (`while(..)`; `for(...)`)

– XSL-Fragment

- ```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
 <xsl:template match="/">
 <H2>Buchbestand</H2>
 <xsl:apply-templates select="INVENTORY/BOOK" />
 </xsl:template>
 <xsl:template match="BOOK">
 Titel:
 <xsl:value-of select="TITLE"/>

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

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

## Grundlagen – *apply-template* XSL-Element

### – Das XSL-*apply-template* Element

- Weist den Browser an, dass für jedes BOOK-Element innerhalb des Wurzelements INVENTORY die Vorlage / das Template verarbeitet werden soll, die mit einem BOOK-Element übereinstimmt

```
– <xsl:template match="/">
 <H2>Buchbestand</H2>
 <xsl:apply-templates select="INVENTORY/BOOK" />
</xsl:template>
```

```
<xsl:template match="BOOK">
 Titel: ...
```



## XML-Dokumente Im Internet Darstellen

### XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

#### Grundlagen – Anzeige mehrerer Elemente : *for-each* XSL-Element

- XML-Fragment

- ..

```
<?xml-stylesheet type="text/xsl" href="XSL apply template.xml"?>
```

```
<INVENTORY>
```

```
 <BOOK>
```

```
 <TITLE>The Adventures of Huckleberry Finn</TITLE>
```

```
 <AUTHOR>
```

```
 <FIRSTNAME>Mark</FIRSTNAME>
```

```
 <LASTNAME>Twain</LASTNAME>
```

```
 </AUTHOR>
```

```
 <BINDING>Taschenbuch</BINDING>
```

```
 <PAGES>336</PAGES>
```

```
 <PRICE>EUR 12,75</PRICE>
```

```
 </BOOK>
```

```
 <BOOK>...
```

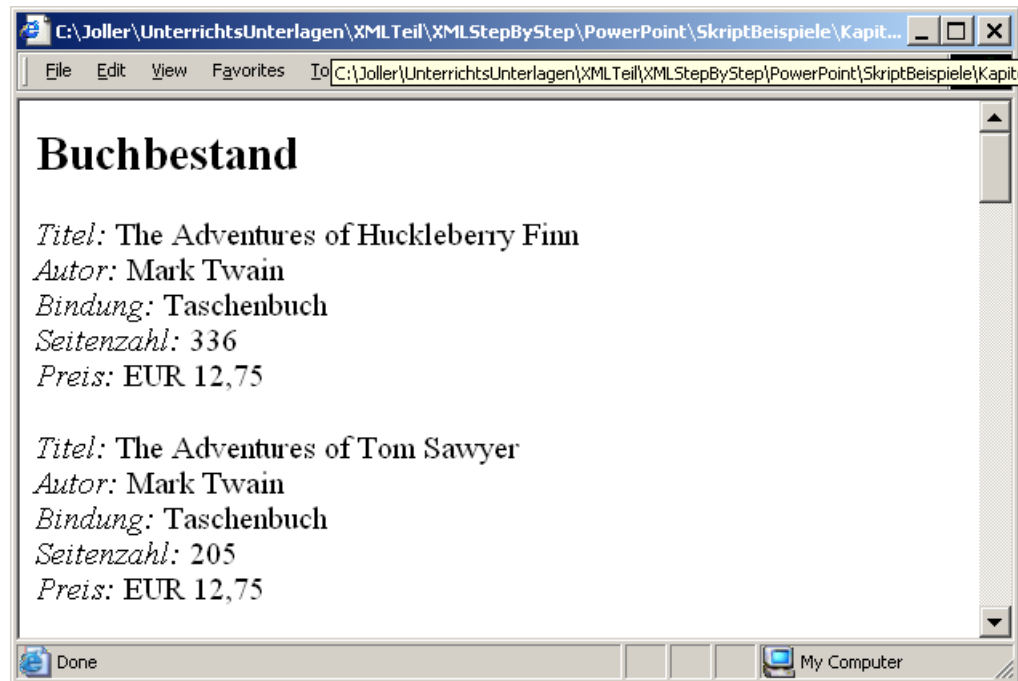
## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

## Grundlagen – Anzeige mehrerer Elemente : *for-each* XSL-Element

### – Ausgabe

- Identisch mit der Ausgabe des *for-each* XSL-Elements



## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

## XML-Daten filtern und sortieren

### – Filter

- Können auf unterschiedliche Art und Weise definiert werden
  - *template*-Element :
    - match**-Attribut
      - » `<xsl:template match=„/“>`
    - *value-of*-Element, *for-each*-Element und *apply-template*-Element:
      - select**-Attribut
        - » `<xsl:for-each select=„INVENTORY/BOOK“>`  
`</xsl:for-each>`
        - » `<xsl:value-of select=„TITLE“/>`
        - » `<xsl:apply-template select=„INVENTORY/BOOK“/>`
  - Zudem können Sie Filter definieren
    - `<xsl:template match=„PfadOperator[Filter]“>`  
*PfadOperator*[*Filter*] bezeichnet man auch als **Muster**
      - » `<xsl:template match=„BOOK[BINDING=„Taschenbuch“]“>`

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

## XML-Daten filtern und sortieren

### – Filter

- Beispielmuster

- match=„ITEM[CD]“

- » Selektiere alle, deren ITEM-Element ein CD-Unterelement besitzt

- match=„SHIRT [COLOR=,rot‘]“

- » Selektiere nur jene Shirts, deren Farbe rot ist

- match=„SHIRT [COLOR!=,rot‘]“

- » Selektiere nur jene Shirts, deren Farbe nicht rot ist

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

## XML-Daten filtern und sortieren

### – Sortieren

- Sortieren macht nur Sinn, wenn mehrere Elemente vorliegen, also in irgend einer Art und Weise eine Schleife
- Folgerichtig wurden in XSL für die „Wiederhol“-Elemente (*for-each*, *apply-template*) ein Attribut **order-by** zum Sortieren definiert (der *order-by* Pfad ist relativ zum *select*-Attribut)

- Beispiele

- `<xsl:for-each`                    **select=**„INVENTORY/BOOK“  
                                          **order-by=**„+AUTHOR/LASTNAME;  
                                                                                          +AUTHOR/FIRSTNAME“>

- » Aufsteigend sortieren, zuerst LASTNAME, dann nach FIRSTNAME

- `<xsl:apply-template`    **select=**„INVENTORY/BOOK“  
                                          **order-by=**„-TITLE“>

- » Absteigend nach TITEL  
(Unterelement von BOOK: relativ zu INVENTORY/BOOK)

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

## XML-Daten filtern und sortieren

- Beispiel

- XSL

- ```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">  
<xsl:template match="/">  
  <H2>Buchbestand</H2>  
  <xsl:for-each  
    select="INVENTORY/BOOK[BINDING='Taschenbuch']"  
    order-by="+AUTHOR/LASTNAME; +AUTHOR/FIRSTNAME">  
    <SPAN STYLE="font-style:italic">Autor: </SPAN>  
    <xsl:value-of select="AUTHOR"/><BR />  
    <SPAN STYLE="font-style:italic">Titel: </SPAN>  
    <xsl:value-of select="TITLE"/><BR />  
    <SPAN STYLE="font-style:italic">Bindung: </SPAN>  
    <xsl:value-of select="BINDING"/><BR />  
    <SPAN STYLE="font-style:italic">Seitenzahl: </SPAN>  
    <xsl:value-of select="PAGES"/><BR />  
    <SPAN STYLE="font-style:italic">Preis: </SPAN>  
    <xsl:value-of select="PRICE"/><P />  
  </xsl:for-each>  
</xsl:template>  
</xsl:stylesheet>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

XML-Daten filtern und sortieren

– Beispiel

- XML

- `<?xml-stylesheet type="text/xsl" href="XSL order by.xsl"?>`

```
<INVENTORY>
```

```
  <BOOK>
```

```
    <TITLE>The Adventures of Huckleberry Finn</TITLE>
```

```
    <AUTHOR>
```

```
      <FIRSTNAME>Mark</FIRSTNAME>
```

```
      <LASTNAME>Twain</LASTNAME>
```

```
    </AUTHOR>
```

```
    <BINDING>Taschenbuch</BINDING>
```

```
    <PAGES>336</PAGES>
```

```
    <PRICE>EUR 12,75</PRICE>
```

```
  </BOOK>
```

```
  <BOOK>
```

```
    <TITLE>The Adventures of Tom Sawyer</TITLE>....
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

XML-Daten filtern und sortieren

– Beispiel

- Ausgabe der selektierten Taschenbücher



XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

XML-Daten filtern und sortieren

- Beispiel

- XSL Alternative

- ```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
 <xsl:template match="/">
 <H2>Buchbestand</H2>
 <xsl:apply-templates select="INVENTORY/BOOK"
 order-by="+AUTHOR/LASTNAME; +AUTHOR/FIRSTNAME"/>
 </xsl:template>
```

```
<xsl:template match="BOOK[BINDING='Taschenbuch']">
 Autor:
 <xsl:value-of select="AUTHOR"/>

 Titel:
 <xsl:value-of select="TITLE"/>

 Bindung:
 <xsl:value-of select="BINDING"/>

 Seitenzahl:
 <xsl:value-of select="PAGES"/>

 Preis:
 <xsl:value-of select="PRICE"/><P />
</xsl:template>
```

## XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

### Auf XML-Attribute zugreifen

- XSL behandelt Attribute von XML-Elementen wie untergeordnete Elemente
- Um auf ein Attribut zugreifen zu können, muss der Name des Attributs mit einem Präfix „@“ versehen werden
- Beispiele
  - `<xsl:for-each select=„INVENTORY/BOOK[@InStock=,ja‘]“>`  
filtert alle BOOK-Elemente aus, deren Attribut *InStock* den Wert ‚ja‘ besitzt.
  - `<xsl:value-of select=„AUTHOR/@Born“/>`  
extrahiert den Wert des *Born*-Attributs des AUTHOR-Elements

# XML-Dokumente Im Internet Darstellen

## XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

### Auf XML-Attribute zugreifen

- Beispiel

- XSL

- ```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">
    <H2>Lieferbare Buchtitel</H2>
    <TABLE BORDER="1" CELLPADDING="5">
      <THEAD>
        <TH>Titel</TH>      <TH>Autor</TH>
        <TH>Bindung</TH>    <TH>Seitenzahl</TH>
        <TH>Preis</TH>      </THEAD>
      <xsl:for-each select="INVENTORY/BOOK[@InStock='ja']">
        <TR ALIGN="CENTER">
          <TD> <xsl:value-of select="TITLE"/> </TD>
          <TD> <xsl:value-of select="AUTHOR"/> <BR/>
            (geboren <xsl:value-of select="AUTHOR/@Born"/>) </TD>
          <TD><xsl:value-of select="BINDING"/> </TD>
          <TD><xsl:value-of select="PAGES"/></TD>
          <TD><xsl:value-of select="PRICE"/></TD>
        </TR>
      </xsl:for-each>
    </TABLE>
  </xsl:template>
</xsl:stylesheet>
```

XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

Auf XML-Attribute zugreifen

– Beispiel

- XML

- `<?xml-stylesheet type="text/xsl" href="XSL Zugriff auf
Attribut.xsl"?>`
`<INVENTORY>`
`<BOOK InStock="ja">`
`<TITLE>The Adventures of Huckleberry Finn</TITLE>`
`<AUTHOR Born="1835">Mark Twain</AUTHOR>`
`<BINDING>Taschenbuch</BINDING>`
`<PAGES>336</PAGES>`
`<PRICE>EUR 12,75</PRICE>`
`</BOOK>`
`<BOOK InStock="nein">`
`<TITLE>Leaves of Grass</TITLE>`
`....`

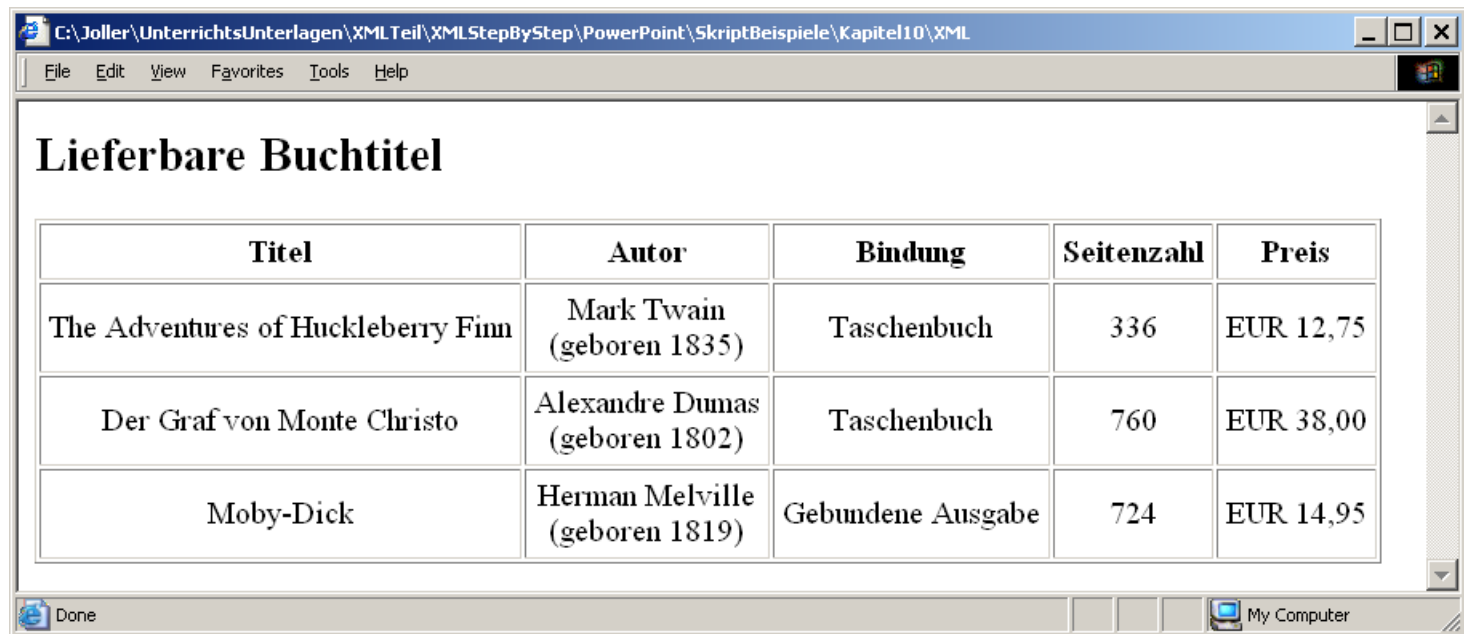
XML-Dokumente Im Internet Darstellen

XML-Dokumente Mithilfe Von XSL-Stylesheets Anzeigen

XML-Daten filtern und sortieren

– Beispiel

- Ausgabe der Bücher, welche an Lager sind



The screenshot shows a web browser window with the title bar 'C:\Joller\Unterrichtsunterlagen\XML Teil\XMLStepByStep\PowerPoint\SkriptBeispiele\Kapitel10\XML'. The browser's address bar is empty, and the menu bar includes 'File', 'Edit', 'View', 'Favorites', 'Tools', and 'Help'. The main content area displays the heading 'Lieferbare Buchtitel' above a table. The table has five columns: 'Titel', 'Autor', 'Bindung', 'Seitenzahl', and 'Preis'. It lists three books: 'The Adventures of Huckleberry Finn' by Mark Twain, 'Der Graf von Monte Christo' by Alexandre Dumas, and 'Moby-Dick' by Herman Melville. The status bar at the bottom shows 'Done' and 'My Computer'.

Titel	Autor	Bindung	Seitenzahl	Preis
The Adventures of Huckleberry Finn	Mark Twain (geboren 1835)	Taschenbuch	336	EUR 12,75
Der Graf von Monte Christo	Alexandre Dumas (geboren 1802)	Taschenbuch	760	EUR 38,00
Moby-Dick	Herman Melville (geboren 1819)	Gebundene Ausgabe	724	EUR 14,95

XML – Eine Einführung

- Links
 - Allgemeine Informationen zu XML
 - Generelle Referez
 - <http://www.w3.org>
 - XML Cover Pages
 - <http://www.oasis-open.org/cover/>
 - XML Anwendungen
 - XML Cover Pages
 - <http://www.oasis-open.org/cover/xml.html#applications>

XML – Eine Einführung

- Links
 - CSS
 - W3C
 - <http://www.w3.org/TR/REC-CSS1>
 - <http://www.w3.org/TR/REC-CSS1>
 - XML Cover Pages
 - <http://www.oasis-open.org/cover/>
 - Namespaces
 - W3C
 - <http://www.w3.org/TR/REC-xml-names>

XML – Eine Einführung

- Links
 - HTML und DHTML
 - W3C
 - <http://www.w3.org/TR/REC-html40>
 - Dokumentobjektmodell (DOM)
 - W3C
 - <http://www.w3.org/TR/REC-DOM-Level1>
Level 2 und Level 3 sind auch bei W3C
 - XSL (Extensible Stylesheet Language)
 - W3C
 - <http://www.w3.org/TR/xsl/>