

XML-Praxis

XSLT, Teil 2 – Übungen

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

Song-Tabelle

- ersetze unordered list durch Tabelle

```
<table>
  <tr><td>Help!</td></tr>
  <tr><td>The Night Before</td></tr>
  ...
</table>
```

Song-Tabelle

- ersetze unordered list durch Tabelle

```
<table>
  <tr><td>Help!</td></tr>
  <tr><td>The Night Before</td></tr>
  ...
</table>
```

- nummeriere die Lieder

```
<table>
  <tr>
    <td>1</td>
    <td>Help!</td>
  </tr>
  ...
</table>
```

Song-Tabelle, cont.

- naive Lösung mit `position()` funktioniert nicht

```
<xsl:template match="songlist">
  <table>
    <xsl:apply-templates/>
  </table>
</xsl:template>
```

```
<xsl:template match="song">
  <tr>
    <td><xsl:value-of select="position()" /></td>
    <td><xsl:apply-templates/></td>
  </tr>
</xsl:template>
```

Song-Tabelle, cont.

- naive Lösung mit `position()` funktioniert nicht

```
<xsl:template match="songlist">
  <table>
    <xsl:apply-templates/>
  </table>
</xsl:template>
```

```
<xsl:template match="song">
  <tr>
    <td><xsl:value-of select="position()" /></td>
    <td><xsl:apply-templates/></td>
  </tr>
</xsl:template>
```

- Problem: Text-Knoten mit whitespace werden auch gezählt

Song-Tabelle, cont.

- 1. Möglichkeit

```
<td><xsl:number count="song" /></td>
```

- [cd2html5a.xsl](#)

- 2. Möglichkeit

```
<xsl:apply-templates select="song" />
```

...

```
<td><xsl:value-of select="position()" /></td>
```

- [cd2html5b.xsl](#)

Erscheinungsjahr

- gib Erscheinungsjahr der CD aus, falls vorhanden
`<h1>The Beatles / Help! (1965)</h1>`

Erscheinungsjahr

- gib Erscheinungsjahr der CD aus, falls vorhanden

```
<h1>The Beatles / Help! (1965)</h1>
```

- im cd-template

```
<xsl:if test="@year">  
  (<xsl:value-of select="@year"/>)  
</xsl:if>
```

Erscheinungsjahr

- gib Erscheinungsjahr der CD aus, falls vorhanden

```
<h1>The Beatles / Help! (1965)</h1>
```

- im cd-template

```
<xsl:if test="@year">  
  (<xsl:value-of select="@year"/>)  
</xsl:if>
```

- noch etwas genauer

```
<xsl:if test="@year!=''">
```

Erscheinungsjahr

- gib Erscheinungsjahr der CD aus, falls vorhanden

```
<h1>The Beatles / Help! (1965)</h1>
```

- im cd-template

```
<xsl:if test="@year">  
  (<xsl:value-of select="@year"/>)  
</xsl:if>
```

- noch etwas genauer

```
<xsl:if test="@year!='' ">
```

- [cd2html6.xsl](#)

Erscheinungsjahr, cont.

- färbe Überschrift abhängig vom Erscheinungsjahr ein

```
<h1 style="color:yellow;">The Beatles / Help! (1965)</h1>
```

- unterscheide vor 1970, 1970–1980, 1980–1990, nach 1990
- verwende Variable

Erscheinungsjahr, cont.

- färbe Überschrift abhängig vom Erscheinungsjahr ein

```
<h1 style="color:yellow;">The Beatles / Help! (1965)</h1>
```

- unterscheide vor 1970, 1970–1980, 1980–1990, nach 1990
- verwende Variable

```
<xsl:variable name="color">  
  <xsl:choose>  
    <xsl:when test="@year >= 1990">  
      <xsl:text>red</xsl:text>  
    </xsl:when>  
    ...  
  </xsl:variable>  
<h1 style="color:{$color};">...</h1>
```

- [cd2html7.xsl](#)

CD-Liste sortieren

- sortiere CDs nach Erscheinungsjahr

CD-Liste sortieren

- sortiere CDs nach Erscheinungsjahr
- pull template, Schleife mit Sortierregel

CD-Liste sortieren

- sortiere CDs nach Erscheinungsjahr
- pull template, Schleife mit Sortierregel

```
<xsl:for-each select="cd">  
  <xsl:sort select="@year"/>  
  <xsl:apply-templates select="."/>  
</xsl:for-each>
```

- [cd2html8a.xsl](#)

CD-Liste sortieren

- sortiere CDs nach Erscheinungsjahr
- pull template, Schleife mit Sortierregel

```
<xsl:for-each select="cd">  
  <xsl:sort select="@year" />  
  <xsl:apply-templates select="." />  
</xsl:for-each>
```

- [cd2html8a.xsl](#)

- Alternative

```
<xsl:apply-templates select="cd">  
  <xsl:sort select="@year" />  
</xsl:apply-templates>
```

- [cd2html8b.xsl](#)

Song-Liste

- gib Song-Liste als Absatz aus, Lieder durch Kommata getrennt
`<p>Help!, The Night Before, ..., Dizzy Miss Lizzy</p>`
- verwende `for-each`-Schleife

Song-Liste

- gib Song-Liste als Absatz aus, Lieder durch Kommata getrennt
`<p>Help!, The Night Before, ..., Dizzy Miss Lizzy</p>`
- verwende `for-each`-Schleife
- Problem: kein Komma am Ende

Song-Liste

- gib Song-Liste als Absatz aus, Lieder durch Kommata getrennt

```
<p>Help!, The Night Before, ..., Dizzy Miss Lizzy</p>
```

- verwende `for-each`-Schleife
- Problem: kein Komma am Ende

```
<xsl:for-each select="song">  
  <xsl:apply-templates/>  
  <xsl:if test="position() != last()">  
    <xsl:text>, </xsl:text>  
  </xsl:if>  
</xsl:for-each>
```

- [cd2html9.xsl](#)