

XML-Praxis

XSL-FO – Übungen

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

XSL-FO Engines

- übersetze `minimal.fo` in PDF-Datei
- XEP von RenderX:

```
$ xep minimal.fo
```
- FOP von Apache:

```
$ fop minimal.fo minimal.pdf
```
- mit PDF-Viewer ansehen
- Empfehlung: `xpdf`, besitzt Reload-Funktion

XSL-FO Engines

- übersetze `minimal.fo` in PDF-Datei
- XEP von RenderX:

```
$ xep minimal.fo
```
- FOP von Apache:

```
$ fop minimal.fo minimal.pdf
```
- mit PDF-Viewer ansehen
- Empfehlung: `xpdf`, besitzt Reload-Funktion
- Unterschiede in der Darstellung? Verarbeitungsgeschwindigkeit?

erste Schritte mit XSL-FO

- definiere Seitenlayout:
 - Seitenränder
 - regions
- setze mehrere Blöcke untereinander
- verändere `space`, `border` und `padding`
- visuelles debugging: `border="solid 1pt green"`
- definiere inline-Bereiche
- teste verschiedene Schriftdefinitionen

XSL-FO mit XSLT erzeugen

- minimales Stylesheet (pro CD ein Block): `cd2fo1.xsl`
- Ausgabe in Datei umlenken

```
$ xsltproc cd2fo1.xsl cd-collection.xml > cd-collection.fo
```

XSL-FO mit XSLT erzeugen

- minimales Stylesheet (pro CD ein Block): `cd2fo1.xsl`
- Ausgabe in Datei umlenken

```
$ xsltproc cd2fo1.xsl cd-collection.xml > cd-collection.fo
```

- erzeuge Überschrift mit Künstler und Titel
- wie bei HTML: doppelte Anzeige verhindern

XSL-FO mit XSLT erzeugen

- minimales Stylesheet (pro CD ein Block): `cd2fo1.xsl`
- Ausgabe in Datei umlenken

```
$ xsltproc cd2fo1.xsl cd-collection.xml > cd-collection.fo
```

- erzeuge Überschrift mit Künstler und Titel
- wie bei HTML: doppelte Anzeige verhindern
- Abstand zum nachfolgenden Block
- jede CD auf eigene Seite

XSL-FO mit XSLT erzeugen, cont.

- Überschrift

```
<xsl:template match="cd">
  <fo:block>
    <fo:block space-after="20pt"
      font="bold 17pt sans-serif"
      text-align-last="center"
      break-before="page">
      <xsl:value-of select="artist"/>
      <xsl:text> / </xsl:text>
      <xsl:value-of select="title"/>
    </fo:block>
    <xsl:apply-templates select="songlist"/>
  </fo:block>
</xsl:template>
```

- [cd2fo2.xsl](#)

Song-Liste

- stelle Song-Liste durch Abfolge von blocks dar
- Bullet: `•`

Song-Liste

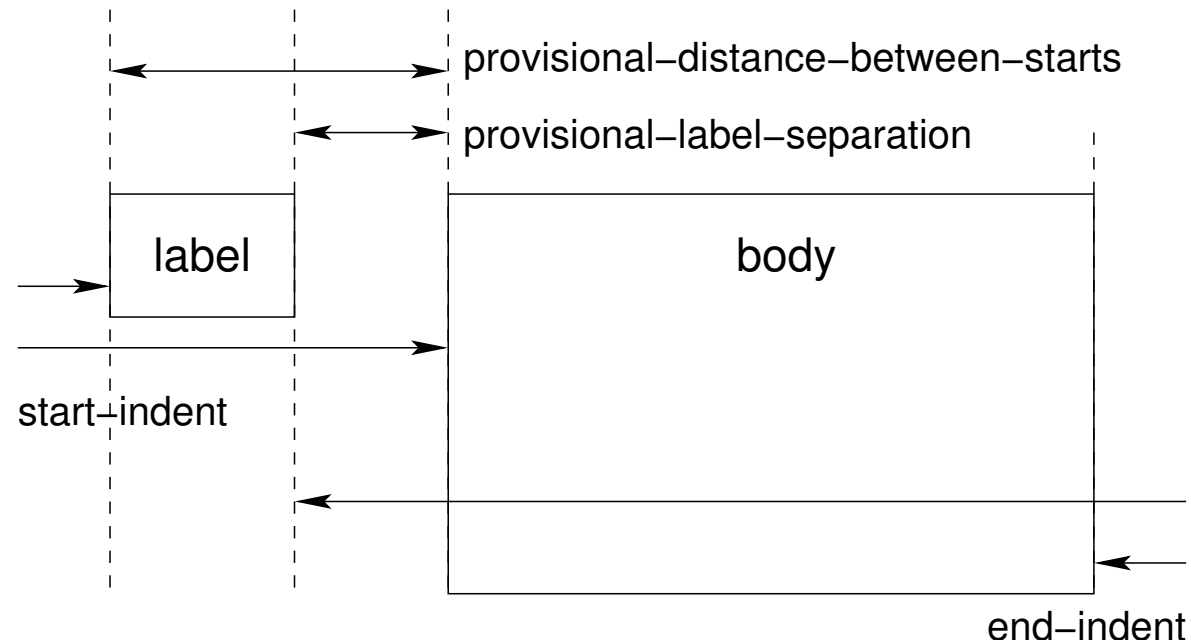
- stelle Song-Liste durch Abfolge von blocks dar
- Bullet: •

```
<xsl:template match="songlist">
  <fo:block space-after="1em">Inhalt:</fo:block>
  <fo:block start-indent="2em">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

```
<xsl:template match="song">
  <fo:block space-after="0.2em">
    <xsl:text>&#x2022; </xsl:text><xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

- [cd2fo3.xsl](#)

Listen



- komplexe Interaktion zwischen Längen
- Funktionen `label-end()` und `body-start()`

Listen, cont.

```
<fo:list-block
  provisional-distance-between-starts="10mm"
  provisional-label-separation="2mm">
  <fo:list-item>
    <fo:list-item-label end-indent="label-end()">
      ...
    </fo:list-item-label>
    <fo:list-item-body start-indent="body-start()">
      ...
    </fo:list-item-body>
  </fo:list-item>
  <fo:list-item>
    ...
  </fo:list-item>
</fo:list-block>
```

Song-Liste als XSL-FO-Liste

- stelle Song-Liste durch `fo:list-block` dar

Song-Liste als XSL-FO-Liste

- stelle Song-Liste durch `fo:list-block` dar
- [cd2fo4.xsl](#)

Seitenzahlen

- definieren Fußzeile

Seitenzahlen

- definieren Fußzeile

```
<fo:region-body margin-bottom="1cm" />
```

```
<fo:region-after extent="1cm" />
```


Seitenzahlen

- definieren Fußzeile

```
<fo:region-body margin-bottom="1cm" />
```

```
<fo:region-after extent="1cm" />
```

- aktuelle Seitenzahl: `<fo:page-number />`
- schreibe Seitenzahl in Fußzeile

Seitenzahlen

- definieren Fußzeile

```
<fo:region-body margin-bottom="1cm" />
```

```
<fo:region-after extent="1cm" />
```

- aktuelle Seitenzahl: `<fo:page-number />`

- schreibe Seitenzahl in Fußzeile

```
<fo:static-content flow-name="xsl-region-after">
```

```
  <fo:block text-align-last="right">
```

```
    <fo:page-number />
```

```
  </fo:block>
```

```
</fo:static-content>
```

- [cd2fo5.xsl](#)