

XML-Praxis

# **XSL-FO**

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

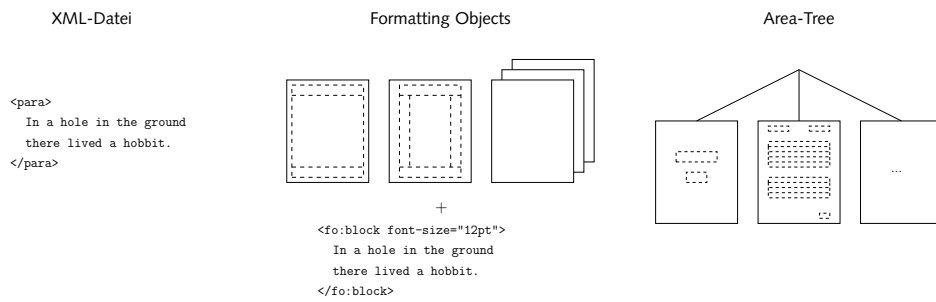
# Übersicht

- XSL Formatting Objects
- Seitendefinitionen
- Blöcke und Inlines

## Eigenschaften von XSL-FO

- zweiter Teil von XSL (neben XSLT)
- XML-Sprache
- beschreibt Seitenlayout und Anordnung von Seiteninhalt
- viele Eigenschaften definierbar (Schriften, Farben, Abstände, ...)
- universelle Beschreibung (Textrichtung, Zeichensätze, ...)
- sollte mit Hilfe von XSLT erzeugt werden

# Arbeitsweise von XSL-FO



- *formatting objects*: page master, blocks, inline elements, ...
- *formatting properties*: font-size, border-color, ...
- *traits*: aus formatting properties abgeleitete area-Eigenschaften

## Aufbau einer FO-Datei

```
<?xml version="1.0"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="dinA4"
      page-width="21cm" page-height="29.7cm">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="dinA4">
    <fo:flow flow-name="xsl-region-body">
      <fo:block font-size="12pt">
        In a hole in the ground there lived a hobbit.
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

## Aufgaben

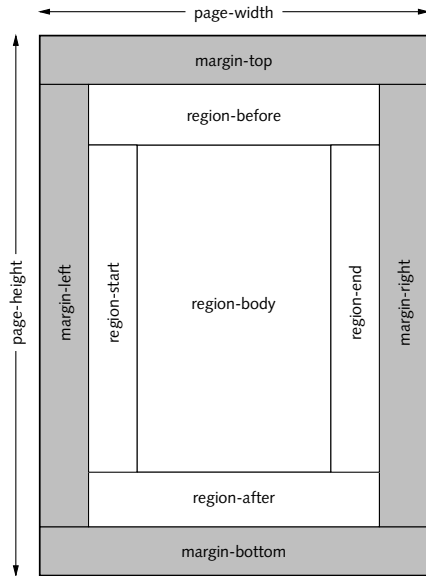
- Übersetze die Datei `minimal.fo` in eine PDF-Datei. Probiere die beiden installierten XSL-FO-Engines aus:

```
$ xep -fo minimal.fo minimal_xep.pdf
$ fop -fo minimal.fo minimal_fop.pdf
```

Sieh Dir die Ergebnisse im Acrobat Reader an. Was für Unterschiede fallen Dir auf?

- Die beiden PDF-Dateien sind im Prinzip identisch. In beiden Dateien wird der Text unmittelbar und ohne Abstand in die linke obere Ecke der DIN-A4-Seite gesetzt. Das Programm `fop` arbeitet etwas schneller als `xep`, dieser Effekt wird bei den späteren Aufgaben noch deutlicher. Allerdings sind in `xep` mehr Teile der XSL-FO-Spezifikation implementiert. Die mit `xep` erstellte Datei enthält einen Hinweis darauf, daß sie mit der akademischen Version des Programms erstellt wurde.

## Seitenaufbau



```
<fo:simple-page-master
  master-name="dinA4"
  page-width="21cm"
  page-height="29.7cm"
  margin-top="1cm"
  margin-bottom="1cm"
  margin-left="1cm"
  margin-right="1cm">
  <fo:region-body
    margin-top="1cm"
    margin-bottom="2cm" />
  <fo:region-before
    extent="1cm" />
  <fo:region-after
    extent="2cm" />
  ...
</fo:simple-page-master>
```

## Aufgaben

- In der FO-Datei `regions.fo` sind die fünf verschiedenen Regionen farblich hervorgehoben. Übersetze die Datei und sieh sie Dir an.
- Verändere die Angaben für die Papiergröße und die Ausdehnung der verschiedenen Regionen. Vergewissere Dich vor allem davon, daß der `region-body` mit den Randregionen überlappen kann, wenn man seine Ränder zu klein wählt.



## Seitenaufbau, cont.

- verschiedene `simple-page-master` nach Bedarf
- können zu `page-sequence-master` zusammengefaßt werden
- komplexe Seitenfolgen:
  - `single-page-master-reference`
  - `repeatable-page-master-reference`
  - `repeatable-page-master-alternatives`
  - `conditional-page-master-reference`
- typische Anwendung: Kapiteleinleitungen, linke/rechte Seiten
- Maximum für Seitenzahlen definierbar

## Materialfluß

- Material wird auf page-sequence verteilt

```
<fo:page-sequence master-reference="dinA4">
```

- Material „fließt“ in bestimmte Region

```
<fo:flow flow-name="xsl-region-body">  
  <fo:block>...</fo:block>  
</fo:flow>
```

- neue Seite (entsprechend sequence), wenn Region gefüllt ist
- statischer Inhalt, der auf jeder Seite erscheint

```
<fo:static-content flow-name="xsl-region-after">  
  <fo:block>Copyright 2002</fo:block>  
</fo:static-content>
```

## XSL-FO mit XSLT erzeugen

- XSL-FO nicht für direkte Bearbeitung geeignet
- stattdessen: FO-Datei mit Hilfe von XSLT erzeugen
- Seitendefinitionen im obersten template
- block- und inline-Elemente in passenden templates erzeugen
- alle verwendeten name spaces definieren:

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">

  <xsl:output method="xml" />
```

## Aufgaben

- Schreibe ein XSLT-Skript `poem2fo.xsl`, das die FO-Anweisungen aus `minimal.fo` erzeugt. Ersetze den statischen Text „In a hole in the ground ...“ durch ein `<xsl:apply-templates/>`. Definiere sinnvolle Seitenränder.
- Übersetze die Datei `gedicht1a.xml` (aus der zweiten Stunde) in eine FO-Datei und diese dann in eine PDF-Datei. Dies kann entweder in zwei Schritten erfolgen

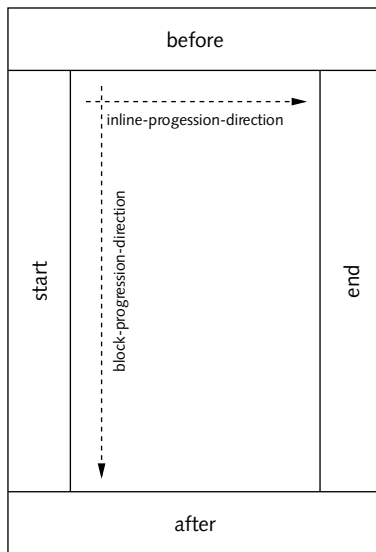
```
$ xsltproc poem2fo.xsl gedicht1a.xml > gedicht1a.fo
$ xep -fo gedicht1a.fo
```

oder man kann den XSLT-Prozessor durch XEP oder FOP aufrufen lassen:

```
$ xep -xml gedicht1a.xml -xsl poem2fo.xsl
$ fop -xml gedicht1a.xml -xsl poem2fo.xsl gedicht1a.pdf
```

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
  <xsl:output method="xml" />
  <xsl:template match="/" >
    <fo:root >
      <fo:layout-master-set >
        <fo:sample-page-master name="dina4"
          page-width="21cm" page-height="29.7cm"
          margin-top="2cm" margin-bottom="2cm"
          margin-left="2cm" margin-right="2cm"
        >
          <fo:region-body />
        </fo:sample-page-master >
      </fo:layout-master-set >
    </fo:root >
  </xsl:template >
  <xsl:template match="*" />
  <xsl:apply-templates />
  </fo:block >
</fo:flow >
</fo:page-sequence >
</fo:root >
</xsl:stylesheet >
```

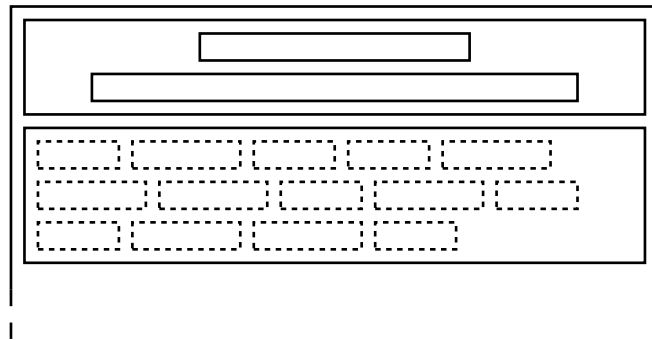
# Areas



- formatting objects → areas
- Seiten-/Zeilenumbrüche
- zwei Arten
  - block areas
  - inline areas
- spezielle Varianten
  - line areas
  - glyph areas

## Blöcke und Inlines

- Blöcke: vertikale Anordnung
- Inlines: horizontale Anordnung



## Blöcke und Inlines, cont.

- Anordnung des Materials:

```
<fo:block>Überschrift</fo:block>
<fo:block>Ein Absatz mit einem
  <fo:inline>hervorgehobenen</fo:inline>
  Wort</fo:block>
```

- Blöcke in Blöcken:

```
<fo:block>
  <fo:block>erster Absatz ...</fo:block>
  <fo:block>zweiter Absatz ...</fo:block>
</fo:block>
```

- formatting properties werden vererbt

# Aufgaben

- Definiere in `poem2fo.xsl` templates für die einzelnen Elemente des Gedichts. Titel und Autor sollen nebeneinander gesetzt werden (der Autor in Klammern), die einzelnen Gedicht-Zeilen untereinander.

```
<fo:page-sequence master-reference="d1nA4">  
  <fo:flow name="xsl-region-body">  
    <fo:block>  
      <xsl:apply-templates select="title|author" />  
    </fo:block>  
    <fo:block>  
      <xsl:apply-templates select="verse" />  
    </fo:block>  
  </fo:flow>  
</fo:page-sequence>  
<xsl:template match="title">  
  <fo:inline><xsl:apply-templates /></fo:inline>  
</xsl:template>  
<xsl:template match="author">  
  <fo:inline><xsl:apply-templates /></fo:inline>  
</xsl:template>  
<xsl:template match="verse">  
  <fo:block><xsl:apply-templates /></fo:block>  
</xsl:template>  
<xsl:template match="line">  
  <fo:block><xsl:apply-templates /></fo:block>  
</xsl:template>
```



## Schriften

- Font-Eigenschaften:

- font-family: Times, Helvetica, ..., serif, sans-serif, ...
- font-style: italic, oblique, backslant
- font-variant: small-caps
- font-weight: bold, bolder, lighter, 100, ..., 900
- font-stretch: wider, narrower, condensed, ...
- font-size: 12pt, 8mm, 75%

- Kurzschreibweise:

```
<fo:block font="bold 15pt serif">  
...
```

## Farben

- Textfarbe:

```
<fo:inline color="red">
```

- Hintergrundfarbe:

```
<fo:block background-color="yellow">
```

- vordefinierte Farben: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow
- RGB-Wert: #ff0055
- RGB-Funktion: rgb(127,255,63)

## Textformatierung

- Unterstreichung:

```
<fo:inline text-decoration="underline">Eigentlich sollte  
  Unterstreichung <fo:inline text-decoration="no-underline">  
  nicht</fo:inline> mehr verwendet werden ...</fo:inline>
```

- ebenfalls: `overline`, `line-through`
- Schatten:

```
<fo:inline text-shadow="red 1pt 1pt">
```

- Buchstaben-/Wortabstände:

```
<fo:inline letter-spacing="5pt">gedehnt</fo:inline>  
<fo:inline word-spacing="5mm">dies ist gesperrt</fo:inline>
```

## Aufgaben

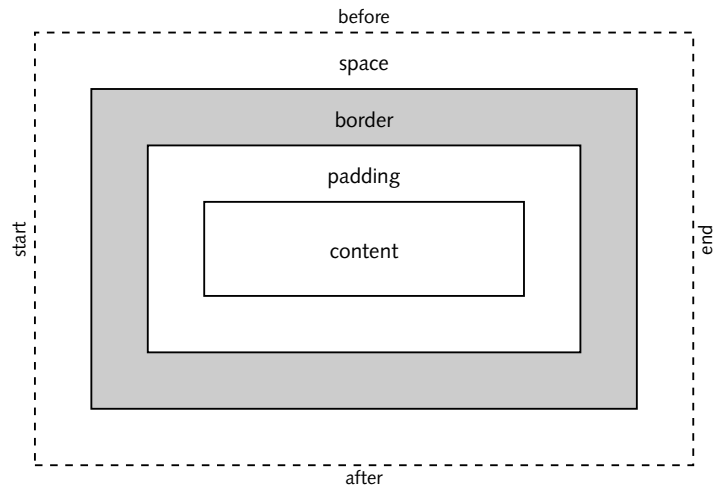
- Füge in `poem2fo.xsl` Anweisungen ein, um Titel und Autor in größerer Schrift zu setzen. Außerdem soll der Name des Autors kursiv dargestellt werden. An welchen Stellen sind die Anweisungen sinnvollerweise einzufügen?
- Hebe einzelne Blöcke und Inlines farblich hervor. Welche Ausdehnung haben die entstehenden areas?

- Blöcke nehmen die gesamte Breite des `region-body` ein, Inlines sind gerade so breit wie ihr Inhalt.

```
<fo:block font-size="24pt"
  <xsl:apply-templates select="title|author" />
</fo:block>
<fo:block font-size="15pt"
  <xsl:apply-templates select="verse" />
</fo:block>
```

- Füge einen weiteren Block um die `verse`-Blöcke:

## Bestandteile einer area



Der Platz um die area (*space*) ist nicht Bestandteil der area selbst. Er ergibt sich bei der Anordnung mehrerer areas durch die FO-Engine.

## padding

- Abstand zwischen content und border
- alle Seiten einzeln definierbar: relative Richtungen ...

```
<fo:block padding-start="2pt" padding-end="3pt">
```

- ... oder absolute Richtungen

```
<fo:block padding-top="2pt" padding-bottom="3pt">
```

- alle Seiten auf einmal:

```
<fo:block padding="2pt">
```

- Kurzformen:

padding="2pt"	<!-- t+l+b+r -->
padding="2pt 4pt"	<!-- t+b l+r -->
padding="2pt 4pt 6pt"	<!-- t l+r b -->
padding="2pt 4pt 6pt 8pt"	<!-- t r b l -->

## border

- Größe, Stil und Farbe:

```
<fo:block border-start-width="1pt"
  border-start-style="solid" border-start-color="blue">
```

- alle Seiten auf einmal:

```
<fo:block border-width="1pt" border-style="ridge"
  border-color="red">
<fo:block border-width="2pt 3pt" border-style="grove"
  border-color="red yellow blue green">
```

- Kurzform:

```
<fo:block border="2pt dotted green">
```

- Kurzform für eine Seite:

```
<fo:block border-top="1pt dashed yellow">
```

## Abstände

- Abstände (*space*) zwischen gleichartigen areas:
  - vertikaler Abstand zwischen block areas
  - horizontaler Abstand zwischen inline areas

- endgültiger Abstand ergibt sich beim Setzen

```
<fo:block space-after="1cm">...</fo:block>  
<fo:block space-before="2cm">...</fo:block>
```

- tatsächlicher Abstand Maximum der beiden Angaben



## Aufgaben

- Füge geeignete Abstände ein:
  - zwischen der Überschrift und der ersten Strophe
  - zwischen den einzelnen Strophen
- Erzeuge einen Rahmen um jede Strophe. Mache Dir den Unterschied zwischen `space` und `padding` klar.
- Erzeuge einen Rahmen um den Titel. Wie verhält sich `padding` bei einem `Inline` im Vergleich zu einem `Block`? Wähle sehr große Werte für das `padding`.

- Bei Blöcken vergrößert `padding` die vertikale Ausdehnung, aber nicht die horizontale, bei `Inlines` ist es genau umgekehrt. Es ist möglich, daß der Rahmen um eine `area` anderes Material auf der Seite überdeckt, obwohl die `areas` selber sich nicht überlappen.

```
<fo:block font-size="24pt" space-after="1em" >
<xsl:apply-templates select="title|author" />
</fo:block>
<fo:block font-size="15pt" >
<xsl:apply-templates select="verse" />
</fo:block>
<xsl:template match="verse" >
<fo:block space-after="1em" >
<xsl:apply-templates />
</fo:block>
</xsl:template>
```

- Abstände:

## Positionierung von Blöcken

- Block nimmt gesamte Breite ein
- `space-start` und `space-end` ohne Einfluß
- Ränder verschieben:

```
<fo:block start-indent="3em" end-indent="3em">
```

## Aufgaben

- Definieren ein template für ein Zitat-Element `quote`.
- Formatiere die Strophen des Gedichts so, daß jede zweite Zeile etwas eingerückt wird:

Wer reitet so spät durch Wind und Nacht?  
Es ist der Vater. Es ist gleich acht.  
Im Arm den Knaben er wohl hält,  
er hält ihn warm, denn er ist erkält'.

```
<xsl:template match="line">
  <fo:block>
    <xsl:if test="position() mod 2 = 0">
      <xsl:attribute name="start-indent">1em</xsl:attribute>
    </xsl:if>
  </xsl:template>
</xsl:template>
<xsl:template match="verse">
  <fo:block space-after="1em"
  <xsl:apply-templates select="line"/>
</fo:block>
</xsl:template>
```

- jede zweite Gedicht-Zeile einrücken:

```
<xsl:template match="quote">
  <fo:block start-indent="3em" end-indent="3em"
  space-before="1em" space-after="1em">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>
```

- Zitat: