



Einführung in XSL-FO

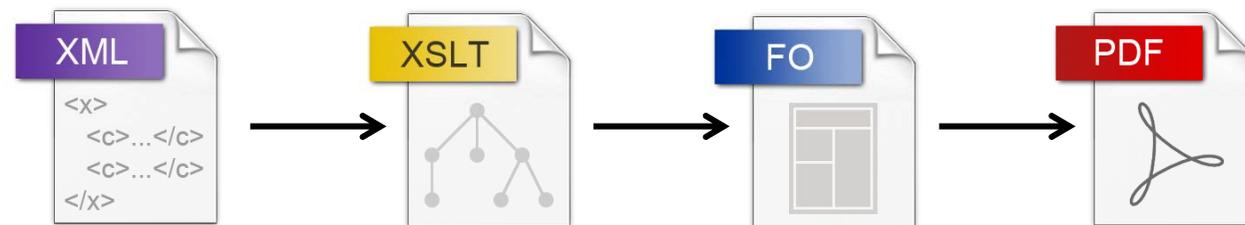
Druckvorlagen erzeugen

Martina Semlak
martina.semlak@uni-graz.at



Überblick

- XSL-FO Grundlagen
- Seitenvorlagen und Seitenaufbau definieren
- Inhalte einfügen
- Seitenfolgenreihenfolgen
- XSLT + XSL-FO
- Ausblick
- Ressourcen





XSL-FO (Formatting Objects)

- Ein FO-Dokument ist eine nach einem W3C-Standard strukturierte XML-Datei, muss also *valide* und *wohlgeformt* sein
- Die Erstellung eines FO-Dokuments erfolgt über ein *XSL-Stylesheet*, das Elemente aus dem *Namensraum*
`xmlns:fo="http://www.w3.org/1999/XSL/Format"` enthält
- XSL-FO ist ein *Teilkonzept* von XSL (XSLT, XPATH, XSL-FO)
- FO-Dokumente enthalten *seitenbezogene Formatierungsanweisungen*, die unabhängig vom Zielformat sind
- Ein FO-Dokument wird von einem *FO-Prozessor* in ein Zielformat umgewandelt (z.B. PDF, RTF oder PostScript)

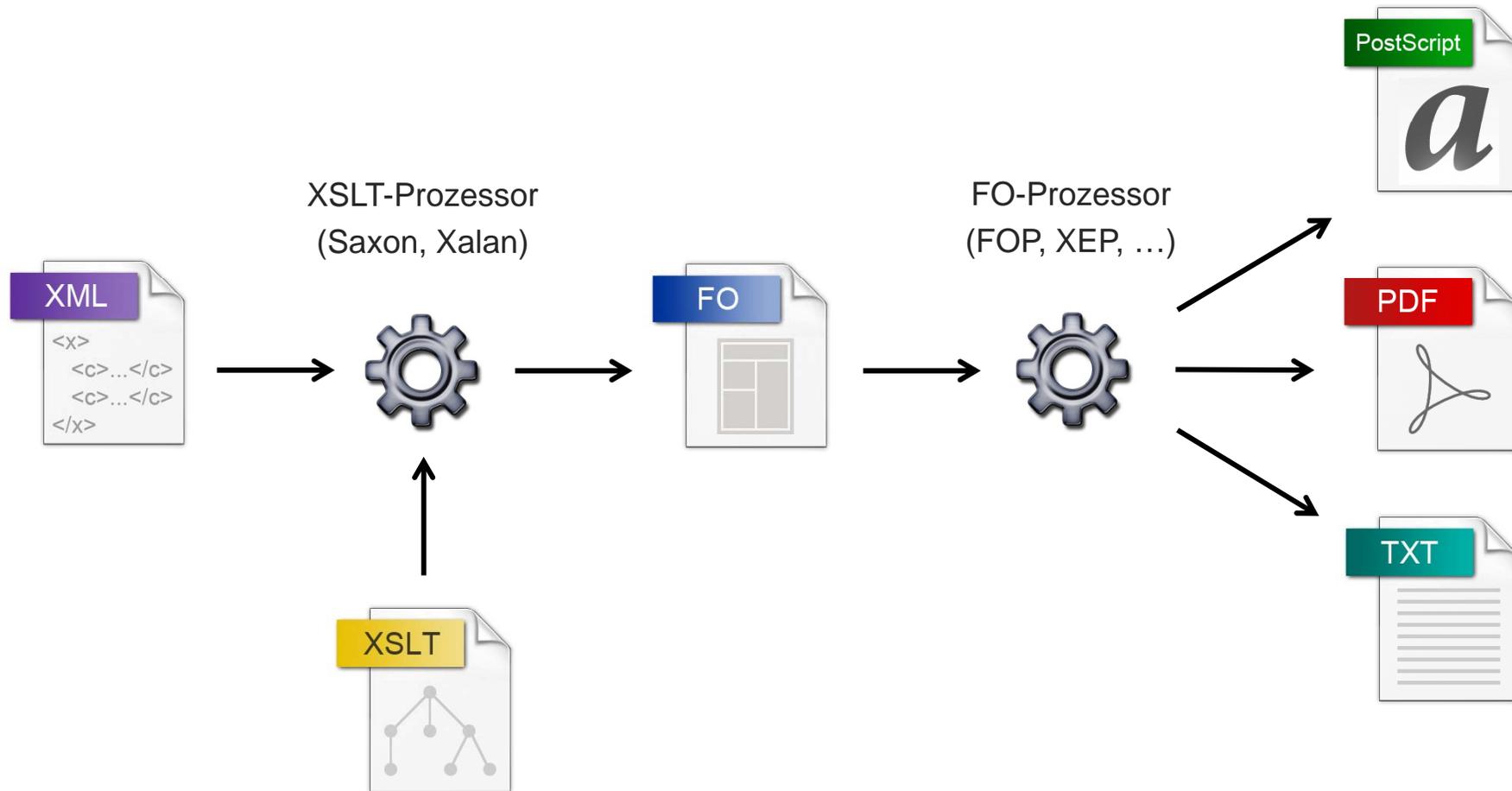


XSL-FO Prozessoren

- Apache FOP (Open Source, in Oxygen integriert)
<http://xml.apache.org/fop/>
- XEP Engine (frei für nichtkommerzielle Nutzung)
<http://www.renderx.com/>
- Antenna House (kostenpflichtig)
<http://www.antennahouse.com/>

- Prozessoren im Vergleich
 - <http://www.data2type.de/xml-xslt-xslfo/xsl-fo/formatierer-vergleich>

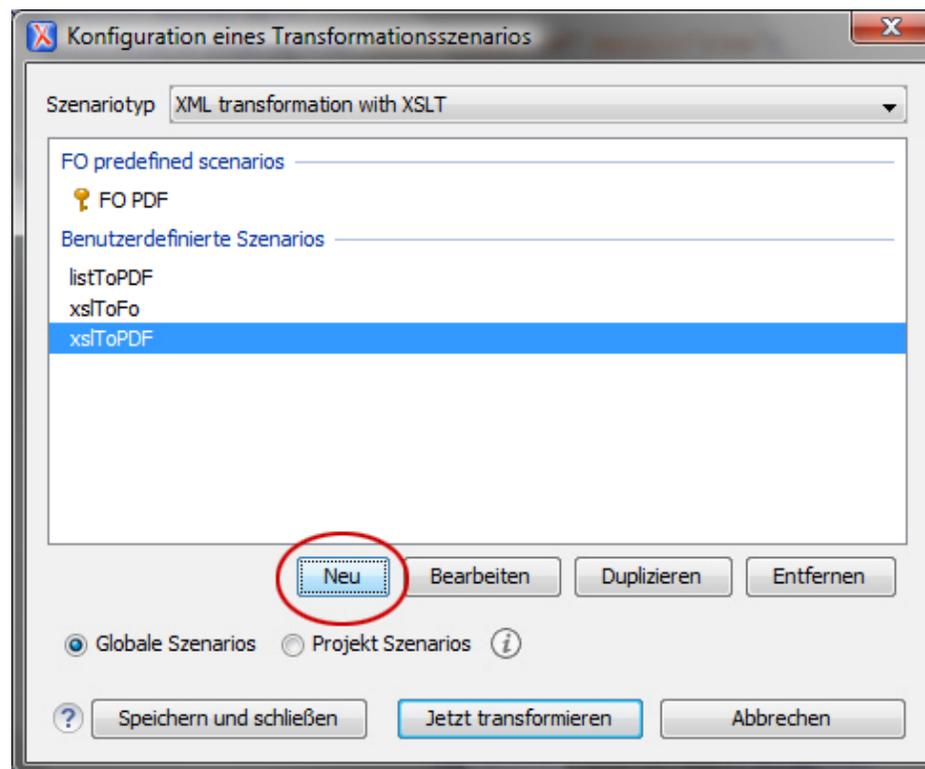
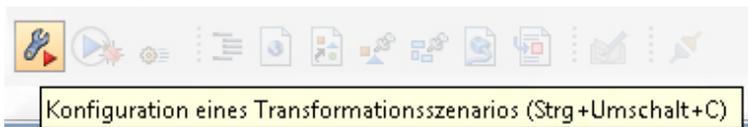
Der Verarbeitungsprozess





Transformationsszenario konfigurieren

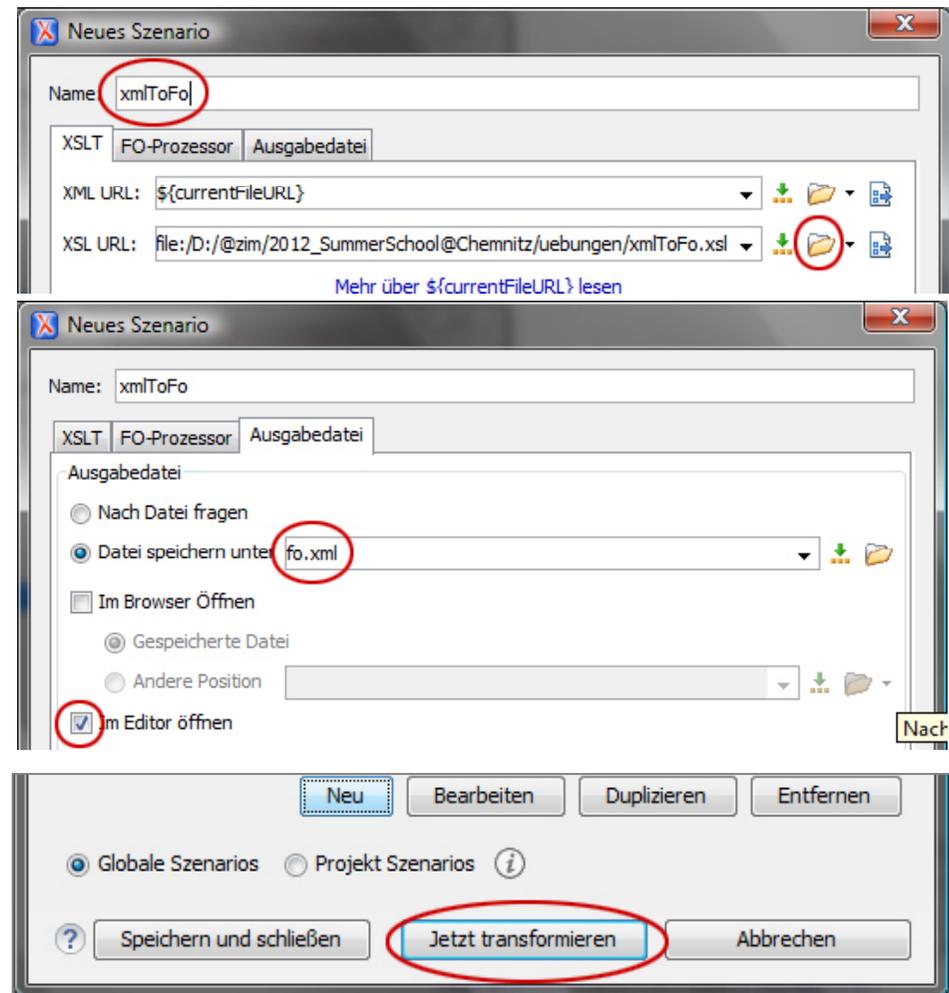
- Konfiguration eines Transformationsszenarios in Oxygen:





Transformation in zwei Schritten: XML to FO

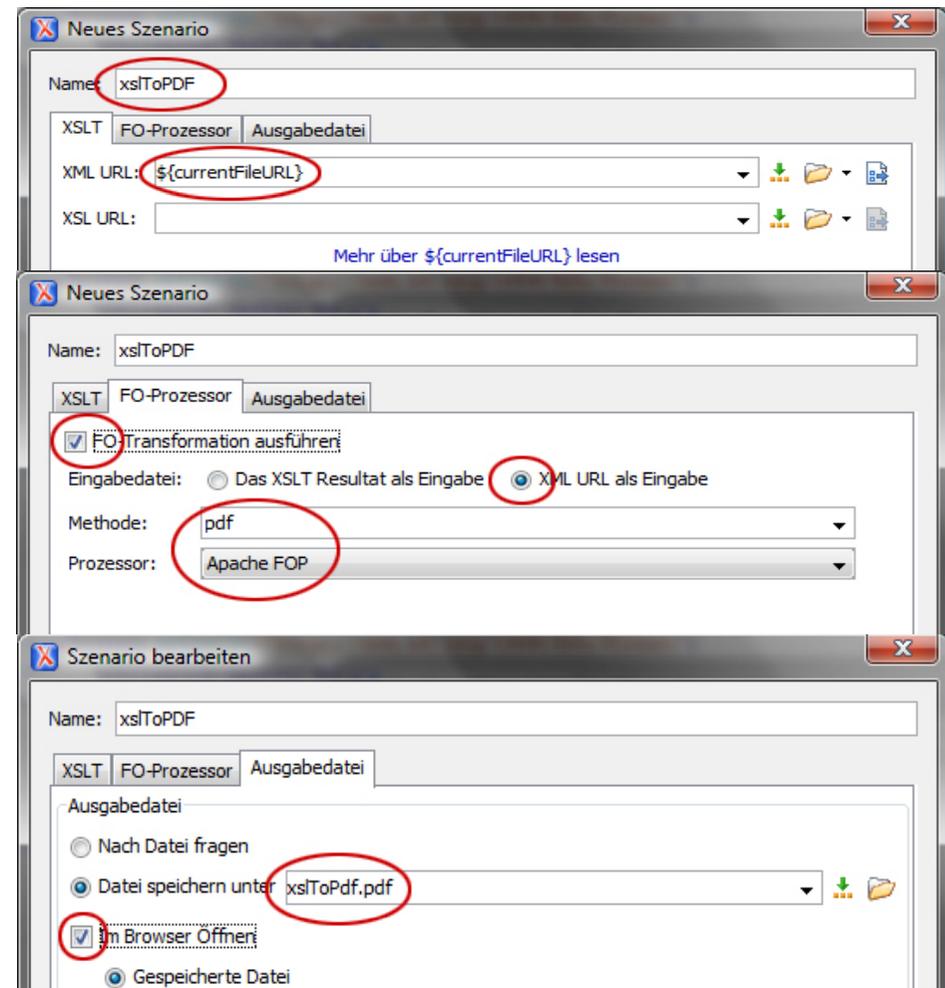
- XSLT:*
 - XML URL `${currentFileURL}`
 - XSL URL: `xmlToFo.xsl`
- Ausgabedatei:*
 - Datei speichern unter: `${cfn}.fo`
(als `.xml` oder `.fo` speichern)
 - Im Editor öffnen
- `${cfn}` = aktueller Dateiname





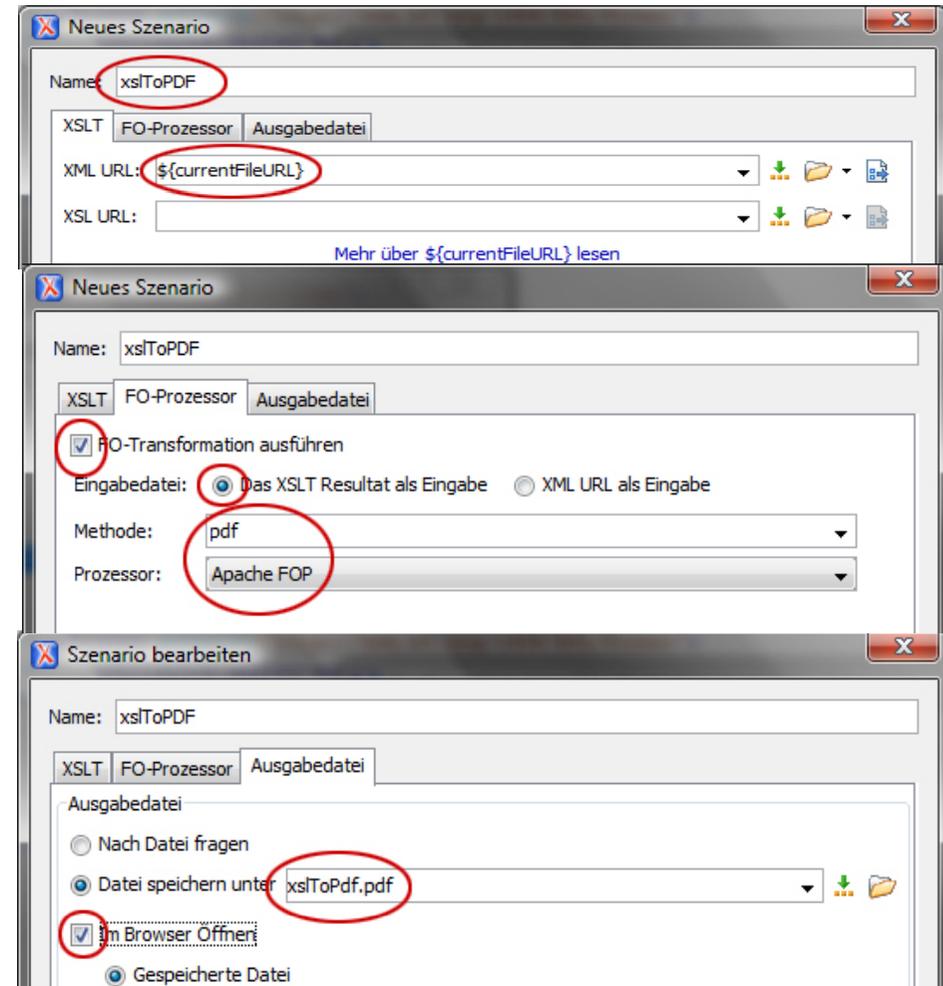
Transformation in zwei Schritten: FO to PDF

- *XSLT:*
 - XML URL `${currentFileURL}`
- *FO-Prozessor:*
 - FO-Transformation ausführen
 - XML URL als Eingabe
 - Methode:pdf
 - Prozessor: Apache FOP
- *Ausgabedatei:*
 - Datei speichern unter: `${cfn}.pdf`
 - Im Browser öffnen



Transformation in einem Schritt: XML to PDF

- Transformationsszenario in einem Schritt XML to PDF
- *XSLT:*
 - XML URL `${currentFileURL}`
 - XSL URL: `xmlToFo.xsl`
- *FO-Prozessor:*
 - FO-Transformation ausführen
 - Das XSLT Resultat als Eingabe
 - Methode:pdf
 - Prozessor: Apache FOP
- *Ausgabedatei:*
 - Datei speichern unter: `${cfn}.pdf`
 - Im Browser öffnen





XSLT und XSL-FO

- Selektierte Knoten werden über XSL-FO-Anweisungen formatiert

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format" version="2.0">
  <xsl:template match="/">
```

```
    <fo:root>
      <fo:layout-master-set>
        <fo:simple-page-master master-name="firstPage"
          page-height="297mm" page-width="210mm" . . .>
          <fo:region-body margin-top="25mm" />
        </fo:simple-page-master>
      </fo:layout-master-set>
      <fo:page-sequence master-reference="firstPage">
        <fo:flow flow-name="xsl-region-body">
          <xsl:apply-templates/>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
```

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

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

```
  <fo:block-container font-size="12pt" margin-left="5mm" margin-bottom="5mm">
```

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

```
  </fo:block-container>
```

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

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

```
  <fo:block>
```

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

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

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

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

XSLT

XML

```
<lg>
  <l>Natur und Kunst</l>
  <l>Und haben sich</l>
  <l>Der Widerwille</l>
  <l>Und beide scheinen</l>
</lg>
<lg>
  <l>Es gilt wohl nur</l>
  <l>Und wenn wir erst</l>
  <l>Mit Geist und Fleiß</l>
  <l>Mag frei Natur</l>
</lg>
```

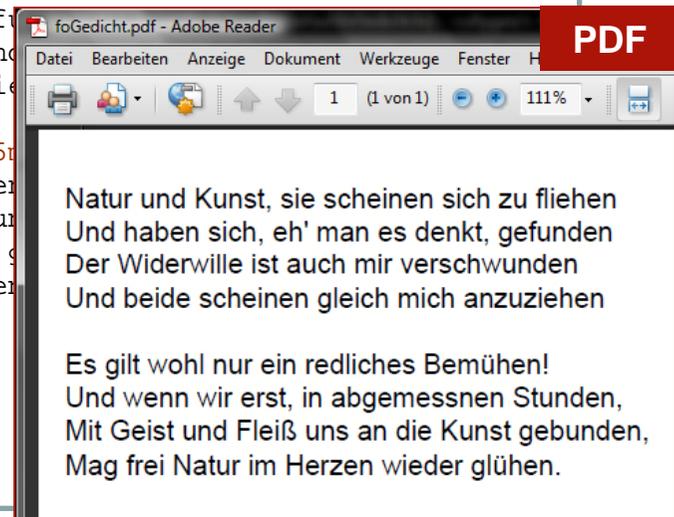


XSLT und XSL-FO

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="firstPage"
      page-height="297mm" page-width="210mm"
      margin-left="12mm" margin-right="12mm"
      margin-top="20mm" margin-bottom="20mm">
      <fo:region-body margin-top="25mm" margin-bottom="20mm" />
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="firstPage">
    <fo:flow flow-name="xsl-region-body">
      <fo:block-container font-size="12pt" margin-left="5mm" margin-bottom="5mm">
        <fo:block>Natur und Kunst, sie scheinen sich zu fliehen</fo:block>
        <fo:block>Und haben sich, eh' man es denkt, gef
        <fo:block>Der Widerwille ist auch mir verschwunden
        <fo:block>Und beide scheinen gleich mich anzuzie
      </fo:block-container>
      <fo:block-container font-size="12pt" margin-left="5mm">
        <fo:block>Es gilt wohl nur ein redliches Bemühen
        <fo:block>Und wenn wir erst, in abgemessnen Stun
        <fo:block>Mit Geist und Fleiß uns an die Kunst gebunden
        <fo:block>Mag frei Natur im Herzen wieder glüher
      </fo:block-container>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

FO

PDF





Übung 1

- Führen Sie in Oxygen eine Transformation in zwei Schritten durch:
 - Transformieren Sie die XML-Datei *uebung1.xml* mittels der XSL-Datei *uebung1.xsl* in eine FO-Datei mit dem Namen *uebung1.fo*
 - Begutachten Sie das Ergebnis
 - Wandeln Sie die FO-Datei in eine PDF-Datei (*uebung1.pdf*) um

- Lösung: *uebung1.xml*, *uebung2.xsl*, *uebung1.fo*, *uebung1.pdf*



Grundgerüst einer FO-Datei

- *fo:root* Wurzelement
- *fo:layout-master-set* Seitenvorlage (Seitengröße, -aufteilung, -abstände)
- *fo:page-sequence* Reihenfolge

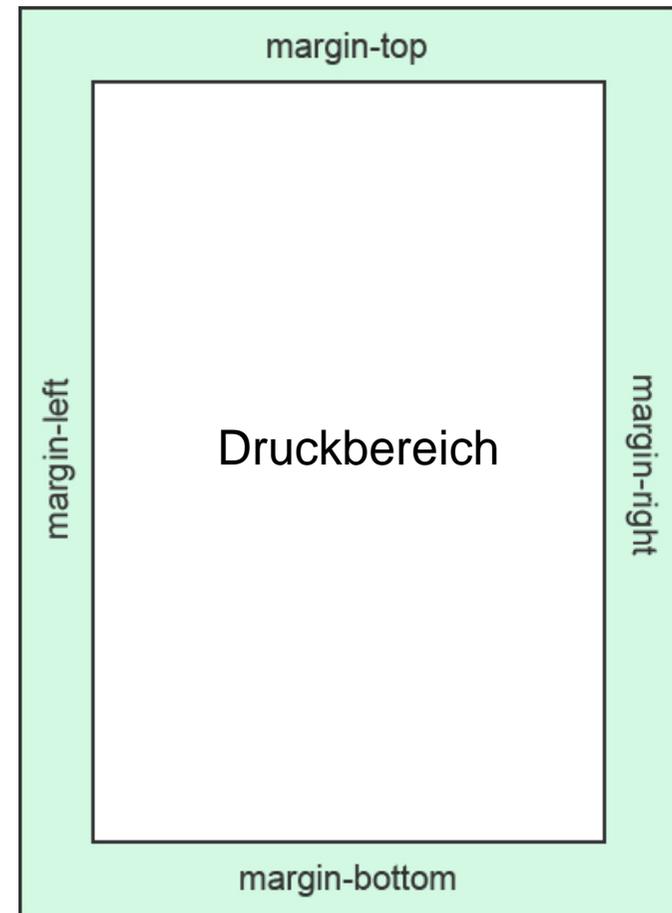
```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="firstPage">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="firstPage">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hello World!</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```



Seitenvorlagen definieren:

- Erfolgt über *fo:simple-page-master*
- Kindelement von *fo:layout-master-set*
- Benennung der Seite über das Attribut *master-name*
- Definition von Seitenabmessungen und -abstände

```
<fo:simple-page-master  
  master-name="A4"  
  page-height="297mm"  
  page-width="210mm"  
  margin-left="12mm"  
  margin-right="12mm"  
  margin-top="20mm"  
  margin-bottom="20mm">
```

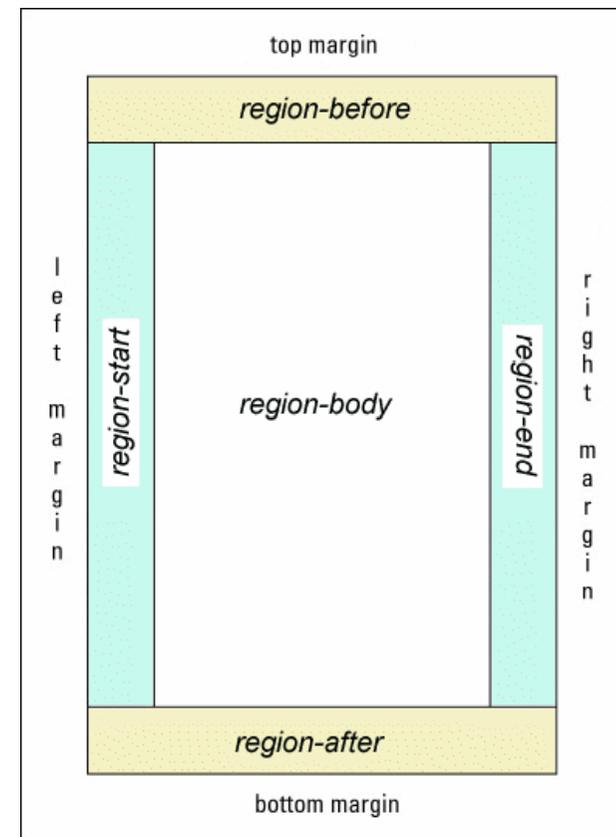




Bereiche definieren: *fo:simple-page-master*

- *fo:region-body* Hauptbereich
- *fo:region-before* Kopfbereich
- *fo:region-after* Fußbereich
- *fo:region-start* Linker Bereich
- *fo:region-end* Rechter Bereich

```
<fo:simple-page-master master-name="A4">  
  <fo:region-body margin="20mm"/>  
  <fo:region-before extent="20mm"/>  
  <fo:region-after extent="20mm"/>  
</fo:simple-page-master>
```





Attribute

- Die Formatierung der Seite(n) und Bereiche erfolgt über Attribute
- Die Attribute und deren Werte sind denen von CSS sehr ähnlich
- Objekte und Attribute
 - http://www.w3schools.com/xslfo/xslfo_reference.asp
- Beispiele:
 - Ränder: *margin*
 - Regionen: *extent* (region-before, region-after); *margin* (region-body);
 - Hintergrund: *background-color*, *background-image*
 - Schrift: *font-family*, *font-weight*, *font-style*, *font-size*
 - Text: *text-align*, *start-indent*, *end-indent*



Anwendung der Seitenvorlage

- Die Anwendung der im *fo:simple-page-master* definierten Seitenvorlage und die Einbettung der Inhalte erfolgt innerhalb von *fo:page-sequence*

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="firstPage">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="firstPage">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>Hello World!</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```



Statische Inhalte und Textflüsse

- Die Seitenvorlage (*fo:page-sequence*) enthält statische und fließende Inhalte
- Statischer Inhalt: *fo:static-content*
 - Wird immer vor fließendem Inhalt notiert
 - Bleibt auf jeder Seite gleich (z.B. Kopf- und Fußzeile)
- Textflüsse: *fo:flow*
 - Inhalte, die dynamisch über mehrere Seiten laufen

```
<fo:page-sequence master-reference="A4">
  <fo:static-content flow-name="xsl-region-before">
    <fo:block border-bottom="0.25pt solid black">IDE Autumn School 2012</fo:block>
  </fo:flow>
  <fo:flow flow-name="xsl-region-body">
    <fo:block>Hello World!</fo:block>
  </fo:flow>
</fo:page-sequence>
```



Inhalte einfügen

- Beispieldatei zu Inhalte einfügen (Blöcke, Listen, Tabellen, ...) als Codeschnipsel: *snippets.fo* (Transformation von FO > PDF, Seite 8)
- ***fo:block***
 - Blockelement, erzeugt Abstand
 - Für Absätze gültig
- ***fo:inline***
 - Inzeilige Formatierung, kann keine Blockelemente enthalten
 - Zur Formatierung einzelner Wörter/Sätze

```
<fo:block font-size="10pt" font-family="Arial, Helvetica, sans" margin-bottom="10mm">
  <fo:inline font-weight="bold" text-align="left">
    IDE Autumn School 2012
  </fo:inline>
  <fo:inline font-style="italic" text-align="center">
    Einführung in XSL-FO
  </fo:inline>
  <fo:inline font-style="normal" text-align="right">
    11.10.2012
  </fo:inline>
</fo:block>
```



Übung 2

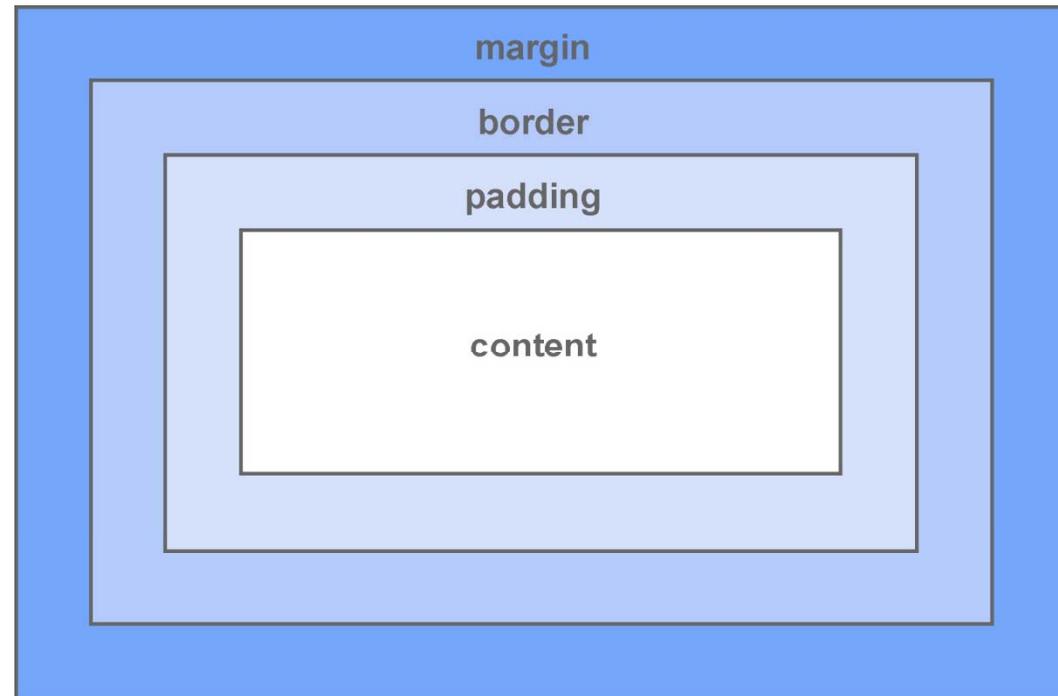
- Erstellen Sie eine Vorlage mit folgenden Eigenschaften:
 - DIN A4 Seite (210x297mm) mit dem Namen „Uebung2“
 - Abstand links und rechts: 12mm
 - Abstand oben und unten: 20mm
 - Eine Kopfzeile mit einer Breite von 25mm und grünem Hintergrund
 - Eine Fußzeile mit einer Breite von 20mm und rotem Hintergrund
 - Wenden Sie die zuvor erstellte Seitenfolge an und fügen Sie folgende Inhalte ein:
 - Kopf- und Fußzeile mit beliebigem Inhalt
 - Textfluss mit 2 Textblöcken und 2 inzeiligen Formatierungen
 - Hinweis: fo:static-content und fo:flow müssen immer Blockelemente (z.B. fo:block) als Kindelement enthalten

- Lösung: *uebung2.fo* (Transformation von FO > PDF, siehe Seite 8)



Little Boxes...

- *margin-top, margin-bottom, margin-left, margin-right*
- *border-top, border-bottom, border-left, border-right*
- *padding-top, padding-bottom, padding-left, padding-right*





Listen und Aufzählungen: *fo:list-block*

- *fo:list-block* ist als Alternative zu *fo:block* zulässig
- Ungeordnete und geordnete Listen sowie Definitionslisten
- Zähler oder Definition wird links freigestellt, Inhalt wird entsprechend eingerückt
- Attribute:
 - *provisional-distance-between-starts*: Breite für das Label

```
<fo:list-block provisional-distance-between-starts="10mm">
```

```
<fo:list-item-body start-indent="body-start()">
```
 - *provisional-label-separation*: Minimalabstand zwischen Label und Body

```
<fo:list-block provisional-label-separation="10mm">
```

```
<fo:list-item-label end-indent="label-end()">
```



```
<fo:list-block start-indent="10mm" provisional-distance-between-starts="10mm"  
    provisional-label-separation="10mm">
```

```
<fo:list-item>  
  <fo:list-item-label end-indent="label-end()">  
    <fo:block>1</fo:block>  
  </fo:list-item-label>  
  <fo:list-item-body start-indent="body-start()">  
    <fo:block>XSLT</fo:block>  
  </fo:list-item-body>  
</fo:list-item>
```

```
<fo:list-item>  
  <fo:list-item-label end-indent="label-end()">  
    <fo:block>2</fo:block>  
  </fo:list-item-label>  
  <fo:list-item-body start-indent="body-start()">  
    <fo:block>XSL-FO</fo:block>  
  </fo:list-item-body>  
</fo:list-item>
```

```
</fo:list-block>
```

Listen	
1	XSLT
2	XSL-FO



Einfache Tabellen

- *fo:table*
Definition einer Tabelle
- *fo:table-body*
Hauptbereich der Tabelle
- *fo:table-row*
Zeile
- *fo:table-cell*
Zelle

```
<fo:block>
  <fo:table>
    <fo:table-body>
      <fo:table-row>
        <fo:table-cell>
          <fo:block> Erste Zelle</fo:block>
        </fo:table-cell>
        <fo:table-cell>
          <fo:block> Zweite Zelle</fo:block>
        </fo:table-cell>
        <fo:table-cell>
          <fo:block> Dritte Zelle</fo:block>
        </fo:table-cell>
      </fo:table-row>
    </fo:table-body>
  </fo:table>
</fo:block>
```

- Komplexe Tabellen: <http://www.data2type.de/xml-xslt-xslfo/xsl-fo/xslfo-einfuehrung/tabellen>



Fußnoten

- Der Fußnotenbereich wird über einen *static-content* der vorgegebenen Seitenfolge eingerichtet
- Fußnoten sind ein Teil des Bereichs *fo:region-body*
- Der Fußnotenbereich wird immer als *xsl-footnote-separator* bezeichnet

```
<fo:static-content flow-name="xsl-footnote-separator">
  <fo:block>
    <fo:leader leader-pattern="rule" rule-thickness="0.5pt" leader-length="33%" />
  </fo:block>
</fo:static-content>
```



Fußnoten

```
<fo:block>Text Text Text Text
<fo:footnote>
  <fo:inline>1</fo:inline>
  <fo:footnote-body>
    <fo:list-block>
      <fo:list-item>
        <fo:list-item-label>
          <fo:block>
            <fo:inline baseline-shift="super">1</fo:inline>
          </fo:block>
        </fo:list-item-label>
        <fo:list-item-body>
          <fo:block>
            <fo:inline baseline-shift="super">Fußnotentext</fo:inline>
          </fo:block>
        </fo:list-item-body>
      </fo:list-item>
    </fo:list-block>
  </fo:footnote-body>
</fo:footnote>
</fo:block>
```

Text Text Text Text Text ¹

¹ Fußnotentext



Paginierung: *initial-page-number*

- Attribut des Elements *fo:page-sequence*

```
<fo:page-sequence initial-page-number="3">  
  ...  
</fo:page-sequence>
```

- Darstellung der Seitenzahlen erfolgt durch das Element *fo:page-number*

```
<fo:static-content flow-name="xsl-region-after">  
  <fo:block>&#x2014; <fo:page-number /> &#x2014;</fo:block>  
</fo:static-content>
```

- Referenzierung von Seitenzahlen (z.B. im Inhaltsverzeichnis) durch *fo:page-number-citation*

```
<fo:block id="Einleitung"> ... </fo:block>  
  
<fo:page-number-citation ref-id="Einleitung" />
```



Marginalien

- Kein spezifisches Konzept für Marginalien in FO
- Erweiterung des *fo:footnote*-Konzepts von Antenna House
- Workaround (für XEP): Marginalien werden als fließende Objekte außerhalb des Blocks positioniert. Problem: linke und rechte Seiten

```
<fo:block intrusion-displace="block">  
  <fo:float float="right">  
    <fo:block margin-left="5mm">Marginalie</fo:block>  
  </fo:float>  
  <fo:block>Textblock</fo:block>  
</fo:block>
```



Übung 3

- Fügen Sie in den Fußzeilenbereich eine Seitenzählung ein
- Versuchen Sie an beliebiger Stelle eine Fußnote einzufügen

- Lösung: *uebung3.fo* (Transformation von FO > PDF, siehe Seite 8)



Grafiken einbinden

- Grafiken und Bilder werden über *fo:external-graphic* eingebunden
- Attribute:
 - *src* Pfad zur Bilddatei
 - *content-height* Höhenausdehnung
auto, scale-to-fit, Maßangabe in cm, %, pt
 - *content-width* Breitenausdehnung
 - Proportionale Anpassung bei Angabe eines Werts

```
<fo:block>
  <fo:inline>
    <fo:external-graphic src="abb1.jpg" />
  </fo:inline>
  <fo:inline>
    <fo:external-graphic src="abb2.jpg" content-width="30mm" />
  </fo:inline>
</fo:block>
```



Ausgewählte Attribute

- Umbrüche steuern
 - *break-before*, *break-after* [column, page, even-page, odd-page, auto]
 - *keep-with-next*, *keep-with-previous*, *keep-together* [Integerwerte, always oder auto]
- Positionierung
 - *position* [absolute, relative], *top*, *bottom*, *left*, *right* [Maßangabe]
- Ausrichtung
 - *reference-orientation* [Gradangaben: 90°, 180°, 270°]



Einzelseiten

- Bei nur einer Seitenfolge wird die Seitenvorlage einer Seitenfolge direkt zugeordnet *fo:simple-page-master* > *fo:page-sequence*

```
<fo:layout-master-set>
  <fo:simple-page-master master-name="A4">
    . . .
  </fo:simple-page-master>
</fo:layout-master-set>
<fo:page-sequence master-reference="A4">
  . . .
</fo:page-sequence>
```



Seitenfolgen einrichten

- Seitensequenzen z.B. für
 - linke Seite, rechte Seite, Kapitelanfang, -ende
 - Erste Seite, restliche Seitenfolge
- Im *fo:layout-page-master* werden die einzelnen Seiten und deren Abstände als *fo:simple-page-master* definiert
- Im *fo:page-sequence-master* wird die Seitenfolge festgelegt
- Im *fo:page-sequence* wird diese Seitenfolge aufgerufen und der Inhalt eingefügt

- Statische Inhalte müssen für jede Seite getrennt angegeben werden, flow-Inhalte werden nur einmal definiert



Seitenfolgen einrichten

```
<fo:layout-master-set>
  <fo:simple-page-master master-name="rechteSeite">
    . . .
  </fo:simple-page-master>
  <fo:simple-page-master master-name="linkeSeite">
    . . .
  </fo:simple-page-master>
  <fo:page-sequence-master master-name="Seitenfolge">
    <fo:repeatable-page-master-alternatives>
      <fo:conditional-page-master-reference master-reference="linkeSeite"
        odd-or-even="even" />
      <fo:conditional-page-master-reference master-reference="rechteSeite"
        odd-or-even="odd" />
    </fo:repeatable-page-master>
  </fo:page-sequence-master>
</fo:layout-master-set>
<fo:page-sequence master-reference="Seitenfolge">
  . . .
</fo:page-sequence>
```

- Beispiel: *seitenfolgen.fo* (Transformation von FO > PDF, siehe Seite 8)



Mehrspaltige Seiten

- Definition im *fo:simple-page-master*
- Im Attribut *column-count* wird die Anzahl der Spalten angegeben, im Attribut *column-gap* wird der Zwischenraum definiert

```
<fo:layout-master-set>  
  <fo:simple-page-master master-name="rechteSeite">  
    <fo:region-body column-count="2" column-gap="10mm" />  
  </fo:simple-page-master>  
</fo:layout-master-set>
```



XSLT und XSL-FO

- Selektierte Knoten werden über XSL-FO-Anweisungen formatiert

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format" version="2.0">
  <xsl:template match="/">
```

XSLT

```
    <fo:root>
      <fo:layout-master-set>
        <fo:simple-page-master master-name="firstPage"
          page-height="297mm" page-width="210mm" . . .>
          <fo:region-body margin-top="25mm" />
        </fo:simple-page-master>
      </fo:layout-master-set>
      <fo:page-sequence master-reference="firstPage">
        <fo:flow flow-name="xsl-region-body">
          <xsl:apply-templates/>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
```

XML

```
<lg>
  <l>Natur und Kunst</l>
  <l>Und haben sich</l>
  <l>Der Widerwille</l>
  <l>Und beide scheinen</l>
</lg>
<lg>
  <l>Es gilt wohl nur</l>
  <l>Und wenn wir erst</l>
  <l>Mit Geist und Fleiß</l>
  <l>Mag frei Natur</l>
</lg>
```

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

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

```
  <fo:block-container font-size="12pt" margin-left="5mm" margin-bottom="5mm">
```

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

```
  </fo:block-container>
```

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

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

```
  <fo:block>
```

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

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

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

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



Formatierungsanweisungen auslagern

- Designanweisungen mit CSS ähnlichen Eigenschaften und Werten
- Zusammenfassung in einem Attributset *xsl:attribute-set*
- Wird zwischen *xsl:stylesheet* und dem ersten *xsl:template* eingerichtet

```
<xsl:attribute-set name="paragraph">  
  <xsl:attribute name="font-size">12pt</xsl:attribute>  
  <xsl:attribute name="line-height">16pt</xsl:attribute>  
  <xsl:attribute name="text-align">left</xsl:attribute>  
</xsl:attribute-set>
```

- Aufruf im Dokument

```
<fo:block xsl:use-attribute-sets="paragraph">Hello World!</fo:block>
```



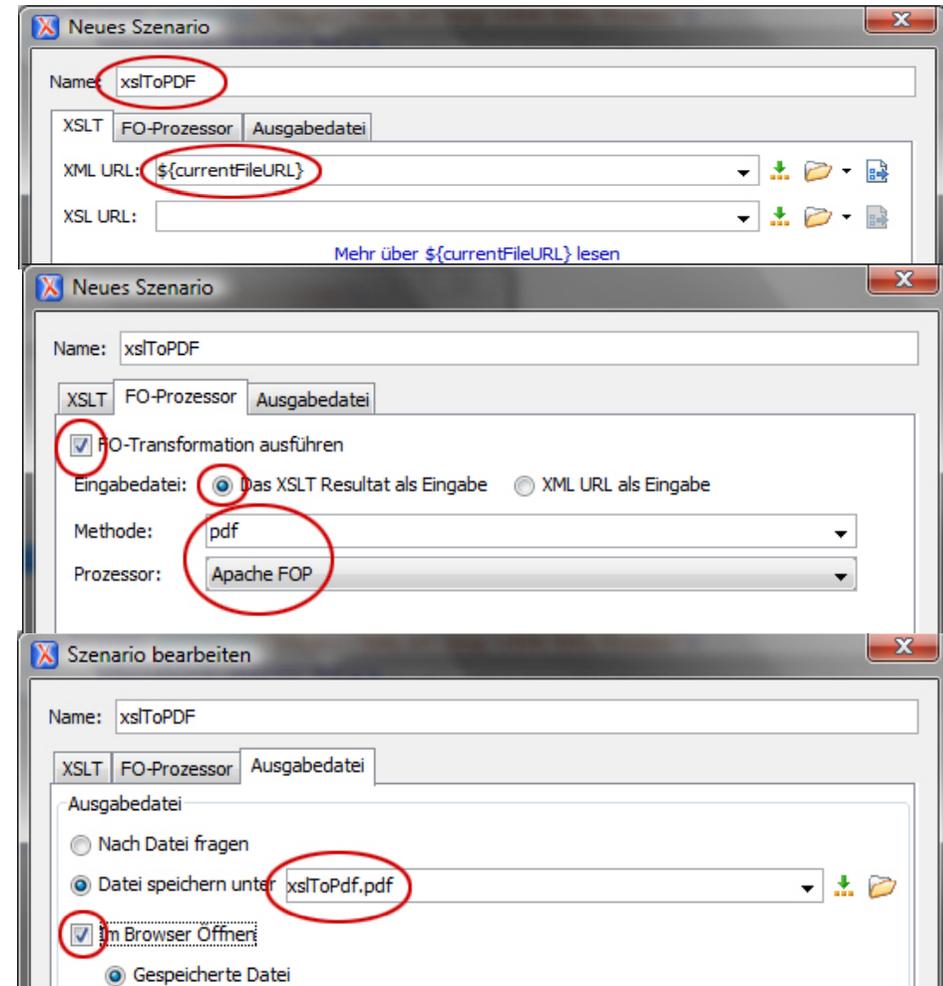
Übung 4

- Erstellen Sie ein XSL-Dokument
- Fügen Sie im Wurzelement den Namensraum für FO-Elemente hinzu
`xmlns:fo="http://www.w3.org/1999/XSL/Format"`
- Fügen Sie ein Template ein, das den Dokumentknoten der XML-Datei adressiert
- Versuchen Sie Inhalte dynamische aus der XML-Datei auszulesen und über FO-Elemente zu formatieren
- Wenn Sie ein TEI-Dokument transformieren, müssen Sie den entsprechenden Namensraum angeben
`xmlns:tei=http://www.tei-c.org/ns/1.0`
- Lösung: *uebung4.xsl* (Transformation von XSL > PDF, siehe S.39)



Transformationsszenario: XML to PDF

- *XSLT:*
 - XML URL `${currentFileURL}`
 - XSL URL: `xmlToFo.xsl`
- *FO-Prozessor:*
 - FO-Transformation ausführen
 - Das XSLT Resultat als Eingabe
 - Methode: pdf
 - Prozessor: Apache FOP
- *Ausgabedatei:*
 - Datei speichern unter: `${cfn}.pdf`
 - Im Browser öffnen
 - `${cfn}` = aktueller Dateiname





Ausblick: XML Print

- Projekthomepage: https://sites.google.com/a/budabe.eu/xmlprint_de/
- Satzprogramm für komplexe wissenschaftliche Texte, z.B. kritische Editionen.
- XSL-FO erweitert = XSL-FO+
- Testversion verfügbar

- Über den *Stileditor* werden Regeln für die Formatierung der XML-Elemente definiert.
- Auf Basis der ausgewählten Attribute wird durch den Start der *Textsatzengine* ein XSLT-Stylesheet generiert, das die XML-Datei in eine XSL-FO+-Datei überführt. Daraus kann dann ein PDF generiert werden.



Ressourcen

Links

- ❑ w3schools.com (<http://www.w3schools.com/xslfo/>)
- ❑ data2type (<http://www.data2type.de/xml-xslt-xslfo/xsl-fo>)
- ❑ <http://www.ibiblio.org/xml/books/bible2/chapters/ch18.html>

Literatur

- ❑ Krüger, Manfred, XSL-FO verstehen und anwenden. XML-Verarbeitung für PDF und Druck, Heidelberg 2006

Mailinglisten

- ❑ www-xsl-fo-request@w3.org (XSL-FO Mailing list)
- ❑ fop-users@xmlgraphics.apache.org (FO Processor Apache FOP Users List)