



# Praktikum zu XML: Übung 12

## XSLT für Bestelldaten



## Organisatorisches



- Arbeitsverzeichnis:  
`~/kurse/xml/12/`
- Dateinamen:  
`09-bestell.xml, 09-bestell.dtd (unverändert)`  
`12-bestell.xsl, 12-bericht.xsl,`  
`12-system.xml, 12-system.xsl, 12-bestell.html`
- Abzugeben:  
`12-system.xsl, 12-bericht.xsl, 12-bestell.xsl`
- Werkzeuge:  

<code>Emacs</code>	<code>Nutzen Sie den XML-Modus!</code>
<code>Mozilla</code>	<code>Zur Kontrolle von HTML-Outputs</code>
<code>Xalan</code>	<code>Der XSLT-Prozessor</code>



- Dateien:
  - 09-bestell.xml, 09-bestell.dtd, 12-system.xml
    - Aus dem Dozentenverzeichnis kopieren
  - 12-system.xsl, 12-bericht.xsl, 12-bestell.xsl
    - Erstellung ist Gegenstand dieser Aufgabe!
- Abgabe:

Dies ist die letzte Praktikumsübung. Die Abgabe erfolgt nicht mehr im Rahmen eines Praktikums-Termins, sondern

**bis spätestens Freitag, den 27.6.2003 18:00 Uhr**

Die Dateien werden –ständig - automatisch eingesammelt!
- Hinweis:
  - Der Umfang dieser Aufgabe ist höher als der von Aufg. 11



- In dieser Aufgabe soll u.a. der Umgang mit Parametern in XSLT geübt werden, daher:
- Hinweis zur Übergabe von Parametern an Xalan:

```
Xalan -p pname pwert myfile.xml mysheet.xsl
```

Achten Sie dabei auf *shell escaping*, etwa:

```
Xalan -p numpar 1 myfile.xml mysheet.xsl
```

Aber:

```
Xalan -p belnr "'X1234'"  
09-bestell.xml 12-bestell.xsl
```



## Übung 12: A - Vorübung



- Erzeugen Sie eine „stylesheet“-Datei **12-system.xml**, die per Textmodus folgende drei XSLT-Prozessorangaben ausgibt:
  - **version**, **vendor**, und **vendor-url**
- Ermitteln Sie die Ausgabewerte zweier XSLT-Prozessoren:
  - A) von Xalan, z.B. mit Aufruf „Xalan 12-system.xml 12-system.xml“
  - B) von Mozilla's eingebautem XSLT-Prozessor, indem Sie **12-system.xml** in Mozilla laden!
- Hinweise
  - Sie benötigen nur ein Element **<xsl:template match="/">**
  - Verwenden Sie **<xsl:text>** zur Ausgabe fester Texte, **<xsl:value-of>** für die Ausgabe von Variablen
  - Die XSLT-Funktion „**system-property()**“ liefert Ihnen die gewünschten Angaben.
  - Tip: Beachten Sie, dass „**vendor**“ etc. ggf. ein *namespace*-Präfix benötigen (z.B. „**xsl**“), und in Hochkommata anzugeben sind.



## Übung 12: A - Vorübung



- Kommentare
  - Beachten Sie die unterschiedlichen Ausgaben!
  - Die Aufgabe ist in 10 Minuten zu schaffen
    - Die Musterlösung enthält nur eine Schablonenregel mit drei Elementen **<xsl:value-of>** sowie zur besseren Lesbarkeit mit einigen Elementen **<xsl:text>**



## Übung 12: B – Ein Bericht mittels XSLT



- Erzeugen Sie eine Tabelle mit einem Bericht zu den in [09-bestell.xml](#) enthaltenen Bestellungen!
- Hinweise und Vorgaben:
  - Legen Sie [12-bericht.xsl](#) an, verwenden Sie den „text“-Modus
  - Der Aufruf „`Xalan 09-bestell.xml 12-bericht.xsl`“ ergebe den Bericht
  - Spaltenüberschriften:
    - Bitte ausgeben: `Best.-Nr.`, `Best.Datum`, `Lieferdatum`, `Positionen`
    - Ausrichtung z.B. einfach mit dem Tabulatorzeichen (`&#9;`)
    - Am einfachsten innerhalb `<xsl:template match="Bestellungen">` anlegen
  - Reihenfolge der Zeilen:
    - Sortieren nach aufsteigendem gewünschtem Lieferdatum!
    - Mit `<xsl:sort>` innerhalb von `<xsl:apply-templates>` an „geeigneter“ Stelle
  - Zeilenausgabe
    - Innerhalb `<xsl:template match="Bestellung">`
    - Mit `<xsl:text>` und `<xsl:value-of>` (und passenden XPath-Ausdrücken!)



## Übung 12: B – Ein Bericht mittels XSLT



- Kommentare:
  - Die Aufgabe ist leichter als sie klingen mag:
  - Die Musterlösung enthält ganze 35 Zeilen Quelltext und 4 Schablonenregeln, davon nur zwei, die nicht-trivial sind.
- Beispiel-Output:

```
werntges@lxl1-01 (~/.kurse/xml/12) $ Xalan 09-bestell.xml 12-bericht.xsl
```

```
Best.-Nr.      | Best.Datum    | Lieferdatum   | Positionen
```

```
=====
```

```
1234568      | 20030416     | 20030530     | 3
AB012345     | 20030528     | 20030615     | 4
1234567      | 20030415     | 20030720     | 2
X54367       | 20030415     | 20030731     | 7
1234569      | 20030418     | 20030920     | 0
```



## Übung 12: C – Bestelldaten lesbar anzeigen



- Teil B ergab, dass Sie Bestellung „X54367“ im Detail sehen möchten. Erzeugen Sie eine HTML-Ausgabe mit dem Inhalt der ausgewählten Bestellung!
- Hinweise und Vorgaben:
  - Legen Sie **12-bestell.xml** an, verwenden Sie den „html“-Modus
  - Übergeben Sie die gewünschte Belegnummer als Parameter!
  - Bsp.: `Xalan -p belnr "X54367" 09-bestell.xml 12-bestell.xml > 12-bestell.html`  
ergebe eine HTML-Datei „12-bestell.html“ mit Bestellung X54367
  - Kontrollieren Sie das Ergebnis durch Laden & Anzeigen der Ausgabedatei mit Mozilla.
- Genereller Aufbau:
  - Analog zu Aufgabe 11, in der ja bereits HTML erzeugt wurde
  - Strukturieren Sie Ihre <template>-Elemente sinnvoll. Bearbeiten Sie z.B. Kopf- und Positionsdaten mit separaten Regeln, überlassen Sie Teilausgaben spezialisierten Regeln
    - etwa für „Datum“ oder „Handelspartner“
  - Orientieren Sie sich am Demo-Output während der Einführung



## Übung 12: C – Bestelldaten lesbar anzeigen



- Allgemein:
  - Die Lösung soll für alle Eingaben von Belegnummern die korrekte Bestellung liefern – mit `<xsl:param>` arbeiten!
- Überschriften:
  - Im Fenstertitel: „**Bestelldaten**“
  - Im Element „h1“: „Bestellung Nr.: xxxxxxx“ (jeweilige Belegnr. einsetzen!)
- Kopfdaten:
  - Handelspartner (alle 2-3 Instanzen):
    - Ausgabe: **Funktion** (etwa: „Besteller“), **IdentArt** („GLN“), **IdentNr**, **Name**
    - Z.B. als kleine HTML-Tabelle erzeugen
  - Datumsangaben
    - Ebenfalls als kleine Tabelle (leicher formatierbar)
    - Fall „**Bestelldatum**“ und „**Gewünschtes Lieferdatum**“ vorsehen



- **Positionsdaten:**
  - Als (neue) Tabelle erzeugen mit Spaltenüberschriften:
    - [Pos](#), [ArtNr](#), [IdentArt](#), [Menge](#), [Mengeneinheit](#), [Beschreibung](#)
  - Viele kleine Regeln anlegen statt wenige komplizierte:
    - Die Ausgabe jedes der 6 gewünschten Element- / Attributwerte einer Bestellposition als eigenes „td“-Element ist besonders einfach, wenn sie jeweils eine eigene kleine Schablonenregel anlegen.
- **Hinweise:**
  - Achten Sie insbesondere auf die Umsetzung des Parameters!
  - Die Musterlösung enthält 117 Quellcodezeilen, verteilt auf 12 Schablonenregeln.
  - Die Datei [12-bestell.html](#) liegt als Orientierungshilfe im Dozentenverzeichnis.