



# Praktikum zur Veranstaltung XML-Technologie: **Übung 11**

XSLT für Bestelldaten  
Umgang mit Parametern  
Benutzung von XSLT-Erweiterungen



## Organisatorisches



- Arbeitsverzeichnis:  
`~/lv/xmltech/11/`
- Dateinamen:  
`08-bestell.xml, 08-bestell.dtd` (unverändert)  
`11-bestell.xsl, 11-bericht.xsl,`  
`11-system.xml, 11-system.xsl, 11-bestell.html`
- Abzugeben:  
`11-system.xsl, 11-bericht.xsl, 11-bestell.xsl`
- Werkzeuge:  
`(X)Emacs` Nutzen Sie den XML-Modus!  
`Firefox` Zur Kontrolle von HTML-Outputs  
`xalan, xsltproc` XSLT-Prozessoren



## Vorbereitungen



- Dateien:
  - 08-bestell.xml, 08-bestell.dtd, 11-system.xml  
Aus dem Dozentenverzeichnis kopieren
  - 11-system.xsl, 11-bericht.xsl, 11-bestell.xsl  
Erstellung ist Gegenstand dieser Aufgabe!
- Abgabe:
  - Die Abgabe erfolgt erst in 14 Tagen
- Hinweis:
  - Der Umfang dieser Aufgabe ist deutlich höher als der von Aufgabe 09 und wird auch stärker gewichtet!



## Übung 11: Hinweise



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

```
xalan -param pname pwert -in myfile.xml  
-xsl mysheet.xsl
```

Achten Sie dabei auf *shell escaping*, etwa:

```
xalan -param numpar 1 -in myfile.xml  
-xsl mysheet.xsl
```

Aber:

```
xalan -param belnr "'x1234'"  
-in 08-bestell.xml -xsl 11-bestell.xsl
```



## Übung 11: A - Vorübung



- Erzeugen Sie eine „stylesheet“-Datei **11-system.xsl**, 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 -in 11-system.xml -xsl 11-system.xsl**“
  - A) von xsltproc, z.B. mit Aufruf  
„**xsltproc 11-system.xml 11-system.xsl**“
  - B) von Mozilla's eingebautem XSLT-Prozessor, indem Sie **11-system.xml** in Mozilla laden!
- Hinweise
  - Sie benötigen nur ein *Tag* **<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.
  - Tipp: Beachten Sie, dass **„vendor“** etc. ggf. ein *namespace*-Präfix benötigen (z.B. **„xsl“**) und in Hochkommata anzugeben sind.



## Übung 11: A - Vorübung



- Kommentare
  - Beachten Sie die unterschiedlichen Ausgaben!
  - Die Aufgabe ist vergleichsweise einfach. Sie 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 11: B – Ein Bericht mittels XSLT



- Erzeugen Sie eine Tabelle mit einem **Bericht** zu den in [08-bestell.xml](#) enthaltenen Bestellungen!
- Hinweise und Vorgaben:
  - Legen Sie [11-bericht.xsl](#) an, verwenden Sie den „[text](#)“-Modus
  - Aufruf „`xalan -in 08-bestell.xml -xsl 11-bericht.xsl`“ erzeuge 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 11: 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@lx1-01 (~/.lv/xmltech/11)$ xalan -in 08-bestell.xml -xsl 11-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 11: C – Bestelldaten lesbar anzeigen



- Teil B ergab, dass Sie Bestellung „X54367“ im Detail sehen möchten.  
**Erzeugen Sie eine XHTML-Ausgabe mit dem Inhalt der ausgewählten Bestellung!**
- Hinweise und Vorgaben:
  - Legen Sie **11-bestell.xsl** an, verwenden Sie den „xml“-Modus
  - Übergeben Sie die gewünschte Belegnummer als Parameter!
  - Bsp.: `xalan -param bealn "X54367" -in 08-bestell.xml -xsl 11-bestell.xsl -out 11-bestell.xhtml`  
    ergebe eine XHTML-Datei „11-bestell.xhtml“ mit Bestellung X54367
  - Kontrollieren Sie das Ergebnis durch Laden & Anzeigen der Ausgabedatei mit Firefox.
- Genereller Aufbau:
  - Analog zu Aufgabe 09, in der ja bereits XHTML 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 11: 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 XHTML-Tabelle erzeugen
  - Datumsangaben  
    Ebenfalls als kleine Tabelle (leichter formatierbar)  
    Fall „**Bestelldatum**“ und „**Gewünschtes Lieferdatum**“ vorsehen



## Übung 11: C – Bestelldaten lesbar anzeigen

---



- **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 [11-bestell.xhtml](#) liegt als Orientierungshilfe im Dozentenverzeichnis.