



Praktikum zu XML: Übung 11

XSLT – Eine HTML-Tabelle



Organisatorisches



- Arbeitsverzeichnis:

`~/kurse/xml/11/`

- Dateinamen:

`11-unicode.xml`

`11-unicode.xsl`

- Abzugeben:

`11-unicode.xsl`

- Werkzeuge:

`Emacs` `Nutzen Sie den XML-Modus!`

`Mozilla` `Zur Kontrolle von HTML-
Outputs`



- Dateien:
 - 11-unicode.xml
 - Aus Ihrer Version 05-unicode.xml ableiten
 - 11-unicode.xsl
 - Erstellung ist Gegenstand dieser Aufgabe!
- Abgabe:
 - Normal, also zu Beginn der nächsten Übung.



- Vorbereitungen:
 - Ändern Sie im Kopf der XML-Datei die PI von der alten CSS- auf die neue XSL-Datei um.
 - Hinweis: Dies betrifft sowohl den Dateinamen als auch den Anwendungstyp.
- Neues Werkzeug: Der XSLT-Prozessor Xalan
 - Pfad: `/usr/local/xml-xalan/bin/Xalan`
 - Verwendung: `Xalan myfile.xml mysheet.xsl`
 - Der Prozessor verwendet die Xerces-Bibliothek
 - Test: Aufruf mit Option `-v` → Versionsangabe, Hilfstexte
 - Sollte 1.4.0 (Xerces 2.1.0) oder aktueller sein
 - Zur Vereinfachung legen Sie ggf. einen neuen alias-Eintrag an...



Übung 11: HTML-Erzeugung mit XSLT



- Aufgabe:
 - Generieren Sie HTML-Ausgabe mit einer Tabelle und einer Überschrift, ähnlich wie in Aufgabe 05 mit CSS.
 - Die in Aufgabe 05 mit CSS erlebten Einschränkungen lassen sich nun überwinden:
 - „Saubere“ Erzeugung einer Überschrift
 - Erzeugung von Spaltenüberschriften ohne Eingriff in die XML-Quelldatei
 - Ausgabe des Wertes von Attribut „TabNr“ in einer eigenen Spalte
 - Frei wählbare Reihenfolge der Spalten



Übung 11: HTML-Erzeugung mit XSLT



- **A: Einfache Version eines Stylesheets**
 - Legen Sie ein leeres Stylesheet an gemäß Vorlesung
 - XML-Deklaration, Kommentar, Elemente stylesheet & output
 - Output-Methode: „html“
 - Schalten Sie die Ausgabe der Textknoten generell ab durch Anlegen einer geeignete (leeren) Schablonenregel.
 - Siehe Vorlesung / Aufg. 10
 - Legen Sie eine Schablonenregel für „/“ an, die „/Codetabelle“ mittels „apply-templates“ durchreicht.
 - Legen Sie eine Schablonenregel für „Codetabelle“ an.
 - Diese soll die HTML-Elemente `html`, `title`, `body` (in dieser Reihenfolge) öffnen und dann mit Inhalt füllen – s.u.
 - Legen Sie innerhalb `body` ein Element `h1` an
 - Füllen Sie es mit einer passenden Überschrift zu Ihrer Tabelle!

Übung 11: HTML-Erzeugung mit XSLT

- **A: Einfache Version eines Stylesheets (Forts.)**
 - Beginnen Sie nun eine HTML-Tabelle (Element `table`)
 - Wahlweise mit Rand - Hinweis auf Attribut „`border`“
 - Erzeugen Sie eine Kopfzeile mit einem Element `tr`, das aus fünf (!) Elementen `th` besteht. Die Überschriftentexte darin sollen lauten (in dieser Reihenfolge!):
 - Zeichen, Unicode, ISO-Code, ISO-Tabelle, Beschreibung
 - Reichen Sie (mit dem passenden XSLT-Element) die Kindelemente von `Codetabelle` weiter.
 - Schließen Sie anschließend alle geöffneten HTML-Elemente.
 - (Forts.)

Übung 11: HTML-Erzeugung mit XSLT

- **A: Einfache Version eines Stylesheets (Forts.)**
 - Erstellen Sie eine Schablonenregel für „`Eintrag`“. Ihre Aufgabe ist die Erzeugung einer Tabellenzeile.
 - Hinweis: HTML-Element `tr` mit 5 Kind-Elementen `td`
 - Befüllen Sie die Elemente `td` mit den Inhalten der entsprechenden Spalten.
 - Verwenden Sie dazu XSLT-Element `value-of`
 - Geben Sie im Attributwert von „`select`“ geeignete XPath-Ausdrücke für die gewünschten Inhalte an
 - Hinweis: Die XPath-Ausdrücke sind relativ einfach.
- Testen Sie Ihr Stylesheet
 - (a) mit Xalan auf Fehlermeldungen; falls ok,
 - (b) indem Sie die XML-Datei mit Mozilla laden.
Erscheint die Tabelle wie erwartet?



Übung 11: HTML-Erzeugung mit XSLT



- **B: Erweiterte Version des Stylesheets** (optional)
 - (1) Kleine XPath-Übung: Geben Sie den Wert von „TabNr“ in eckigen Klammern [..] aus.
 - (2) Befüllen Sie die Zellen, für die kein ISO-Code vorliegt, mit Bindestrichen „-“
Hinweis: XSLT-Elemente [choose](#), [when](#), [otherwise](#)
 - (3) Sortieren Sie die Tabellenzeilen aufsteigend nach dem Wert der Elemente [Unicode](#)
Hinweis: XSLT-Element [sort](#) mit Attribut „[select](#)“ als Unterelement von [apply-templates](#)
- Testen Sie Ihr *Stylesheet* nach jedem Teilschritt wie in Teil A!



Übung 11: Abschließende Hinweise



- Zusätzlich zu den Informationen aus der Vorlesung stehen Ihnen zur Verfügung:
 - Die Beispieldateien aus Aufgabe 10
 - Die XSLT-Spezifikationen
 - SelfHTML 8.0 – darin sind auch alle XSLT-Elemente mitsamt Code-Beispielen verzeichnet.
 - Element [sort](#) war noch nicht behandelt. Schlagen Sie insbesondere dieses Element in den Spezifikationen bzw. in SelfHTML nach!