



Praktikum zur Veranstaltung XML-Technologie: **Übung 09**

XSLT – Eine HTML-Tabelle

- Arbeitsverzeichnis:
`~/lv/xmltech/09/`
- Dateinamen:
`09-unicode.xml`
`09-unicode.xsl`
- Abzugeben:
`09-unicode.xsl`
- Werkzeuge:
`(X)Emacs` Nutzen Sie den XML-Modus!
`Firefox` Zur Kontrolle der HTML-Outputs
`xalan` Der XSLT-Prozessor (optional)



- Dateien:
 - 09-unicode.xml
Aus Ihrer Version 04-unicode.xml ableiten
 - 09-unicode.xsl
Erstellung ist Gegenstand dieser Aufgabe!
- Abgabe:
 - Normal, also zu Beginn der nächsten Übung.



Übung 09: HTML-Erzeugung mit XSLT



- 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
 - Verwendung (bei XHTML-Output):

```
xalan -in myfile.xml -xsl mysheet.xsl -xml  
-out myfile.xhtml -validate
```
 - Der Prozessor verwendet die Xerces-Bibliothek



Übung 09: HTML-Erzeugung mit XSLT



- Aufgabe:
 - Generieren Sie XHTML-Ausgabe mit einer Tabelle und einer Überschrift, ähnlich wie in Aufgabe 04 mit CSS.
 - Die in Aufgabe 04 mit CSS erlebten Einschränkungen lassen sich nun aber überwinden:
 - „Saubere“ Erzeugung einer **Tabellenüberschrift**
 - Erzeugung von **Spaltenüberschriften** ohne Eingriff in die XML-Quelldatei
 - Ausgabe des Wertes von **Attribut** „TabNr“ **in einer eigenen Spalte**
 - Frei wählbare Reihenfolge** der Spalten
- Auflage: „Saubere“ arbeiten
 - Erzeugen Sie XHTML 1.1-Code
 - Wählen Sie den XHTML-Namensraum als Default-Namensraum Ihres Stylesheets!
 - Validieren Sie Ihren Output mittels <http://validator.w3.org>



- **A: Einfache Version eines Stylesheets**

- Legen Sie ein leeres Stylesheet an gemäß Vorlesung
XML-Deklaration, Kommentar, Elemente „stylesheet“ & „output“
Output-Methode: „xml“
- Schalten Sie die Ausgabe der Textknoten generell ab
durch Anlegen einer geeignete (leeren) Schablonenregel.
Siehe Vorlesung
- 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`, `head`, `head/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!



- **A: Einfache Version eines Stylesheets (Forts.)**
 - Beginnen Sie nun eine XHTML-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 XHTML-Elemente. (Forts.)



- **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.



Übung 09: HTML-Erzeugung mit XSLT



- **A: Einfache Version eines Stylesheets (Forts.)**

Testen Sie nun Ihr Stylesheet

- (a) mit Xalan auf Fehlermeldungen;
falls ok:
- (b) Lassen Sie den XHTML-Output des Xalan auf Gültigkeit im Sinne von xhtml1 1.dtd prüfen mit dem Internetwerkzeug unter <http://validator.w3.org>
- (c) Laden Sie die XML-Datei mit dem Firefox-Browser.

Erscheint die Tabelle wie erwartet?



Übung 09: 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 A!



Übung 09: Abschließende Hinweise



- Zusätzlich zu den Informationen aus der Vorlesung stehen Ihnen zur Verfügung:
 - Die Beispieldateien aus der Vorlesung
 - Ein XHTML 1.1-Grundgerüst (nächste Folie)
 - Die XSLT-Spezifikationen
 - SelfHTML 8.1 – 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!



- Beispiel für XHTML 1.1-Dokumente

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<?xml-stylesheet href="hello.css" type="text/css"?>
<html
  xmlns="http://www.w3.org/1999/xhtml" xml:lang="de">
  <head>
    <title>Kleines XHTML 1.1-Beispiel</title>
  </head>
  <body>
    <p>Hallo, Welt!</p>
    <!-- Hier ergänzen! -->
  </body>
</html>
```