

# Einführung in XML

## Konzepte, Arbeitsabläufe

Günter Partosch, Mai 2003 / Februar 2004

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Inhalt](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

## Zusammenfassung:

*Das Tutorium „Einführung in XML“ richtet sich in erster Linie an Einsteiger in den Themenkomplex „Extensible Markup Language (XML)“. Vorkenntnisse sind nicht erforderlich.*

**Extensible Markup Language** ist ein einfaches, sehr flexibles und von SGML abgeleitetes (Meta-)Text-Format, das eine zunehmende Rolle bei Austausch und Archivierung von Textdaten spielt.

*Im Tutorium werden zunächst die zugrunde liegenden Konzepte gezeigt; im folgenden wird dann anhand von kleinen Beispielen demonstriert, wie XML-Dokumente erstellt, bearbeitet, konvertiert und präsentiert werden können.*

- Die jeweils neueste Version des Tutoriums finden Sie unter <http://www.uni-giessen.de/partosch/xml/kurs2/>.
- Beim Aussehen der Folien habe ich mich an *MAPS bibliografie* von Hans Hagen angelehnt.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

# Inhalt

1	Ausgangsposition und Problemstellung	5
2	Markup	7
3	SGML	9
3.1	Dokumenttypen	10
3.2	Deskriptives Markup	11
3.3	Aufbau eines SGML-Dokuments	12
3.4	HTML	13
4	XML	14
4.1	Gründe für XML	15
4.2	Entwurfsziele für XML	17
4.3	XML-Konzepte	18
4.3.1	„Wohlgeformte“ und gültige XML-Dateien	19
4.3.2	XML-Deklaration ( <i>XML declaration</i> )	21
4.3.3	Dokument-Typdeklaration ( <i>document type declaration</i> )	22
4.3.4	Element-Deklaration	23
4.3.5	Attribut-Deklaration	24
4.3.6	Entity-Deklaration	24

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

5	XML – Begleitende Konzepte (Auswahl)	25
6	Standardisierte XML-Applikationen	29
6.1	XHTML	31
6.2	MathML	35
7	Arbeitsabläufe – Überblick	37
7.1	XML-Werkzeuge und -Software	38
7.2	Erstellen von XML-Dateien	39
7.3	Darstellen von XML-Dateien	40
7.3.1	Direkte Darstellung von XML-Dateien	41
7.3.2	Darstellen von PDF-Dateien	44
7.3.3	Anmerkungen zu PDF	45
7.4	Konvertieren von XML-Dateien	46
7.4.1	Direkte Konvertierung	47
7.4.2	Konvertierung nach HTML bzw. Zwischenformaten	48
7.4.3	Konvertierung der Zwischenformate	51
7.4.4	Tool-Ketten	52
7.4.5	Warum sollte hier T <sub>E</sub> X eine Rolle spielen?	55
7.5	Resümee	56
8	Arbeitsabläufe – Beispiele	57
8.1	Gedichtsammlung	58
8.1.1	Erstellen der XML-Datei	59

Ausgangsposition und  
 Markup  
 SGML  
 XML  
 XML – Begleitende Konzepte  
 Standardisierte XML-Applikationen  
 Arbeitsabläufe – Überblick  
 Arbeitsabläufe – Beispiele  
 Informationsquellen (Literatur)  
 URLs  
 Nachtrag: XSL Formate

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

8.1.2	Darstellen der XML-Datei	64
8.2	Tabelle	67
8.2.1	Erstellen der XML-Datei	68
8.2.2	Darstellen der XML-Datei	68
8.2.3	Umformen durch spezialisiertes Programm	68
8.2.4	Darstellen mittels CSS-Datei und Browser	69
8.2.5	Transformation in eine HTML-Datei mittels XSL(T) und Browser	69
8.2.6	Transformation in eine HTML-Datei (mit Sortierung)	70
8.2.7	Transformation der XML-Datei in eine L <sup>A</sup> T <sub>E</sub> X-Datei	71
9	Informationsquellen (Auswahl)	72
9.1	Usenet-News-Gruppen und Diskussionslisten	73
9.2	Anleitungen im Netz	74
9.3	Bücher (Auswahl)	75
10	URLs	77
11	Nachtrag: XSL Formatting Objects	84

Ausgangsposition und  
 Markup  
 SGML  
 XML  
 XML – Begleitende  
 Standardisierte XML  
 Arbeitsabläufe – Über  
 Arbeitsabläufe – Bei  
 Informationsquellen  
 URLs  
 Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

# 1 Ausgangsposition und Problemstellung

## Wunsch

Ein Text kann mit angemessenem Aufwand weltweit

- *eingesetzt*,
- *ausgetauscht* und *verarbeitet* und
- leicht an verschiedene (Rand-)Bedingungen *angepasst* werden.

Ausgangsposition und

Markup

SGML

XML

XML – Begleitende

Standardisierte XML

Arbeitsabläufe – Übe

Arbeitsabläufe – Beis

Informationsquellen

URLs

Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## Notwendige Anforderungen

- *international*: Der Text ist weltweit lesbar und interpretierbar.
- *plattformunabhängig*: Die Verarbeitung des Textes ist nicht abhängig von einem bestimmten Rechnertyp oder Betriebssystem.
- *applikationsunabhängig*: Firmenspezifische Software darf nicht bestimmen, was lesbar/verarbeitbar ist.
- *alterungsbeständig*: Der Text ist auch nach Jahren noch verarbeitbar.
- *ausgebbar in verschiedenen Formaten*

Die meisten Anforderungen lassen sich vergleichsweise unaufwändig und allgemein durch eine innere Strukturierung der Texte verwirklichen, z. B. durch ein **logisches/deskriptives Markup**.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Beis  
Informationsquellen (I  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

## 2 Markup

### Inhaltliches Markup (Deskriptives Markup)

- Man beschreibt, welche *inhaltliche Bedeutung* die einzelnen Textteile haben.
- Man *beschreibt nicht*, in welchem Font, in welcher Größe, in welcher Ausprägung usw. ein Textabschnitt präsentiert werden soll.
- Die *tatsächliche Realisierung geschieht an anderer Stelle*, z. B. durch bestimmte Einstellungen des verarbeitenden Programms bzw. durch Style-Sheets.

Das ist `<em>hervorgehobener</em>` Text.

oder

Das ist `\emph{hervorgehobener}` Text.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch



## Physisches Markup (Präsentations-Markup)

- Verschiedene Textteile sind mit verschiedenen Zeichen-/Absatz-Formatierungseigenschaften versehen.
- Beginn/Ende solcher Textabschnitte sind gekennzeichnet; z. B. durch Umschalt-Sequenzen oder spezielle Konstrukte:
  - zu wenig variabel, wenn Formatierungen nicht darstellbar sind oder geändert werden sollen
  - inhaltliche Struktur nicht erkennbar/erschließbar

Das ist `<i>kursiver Text</i>`.

oder

Das ist `{\it kursiver Text}`.

## SGML und XML

SGML und XML sind *Regelwerke für das Vereinbaren und Anwenden eigener Markup-Sprachen* für deskriptives Markup.

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

# 3 SGML

3.1	Dokumenttypen	10
3.2	Deskriptives Markup	11
3.3	Aufbau eines SGML-Dokuments	12
3.4	HTML	13

---

Mit SGML (Standard Generalized Markup Language) wurde erstmals ein *Regelwerk zum Vereinbaren und Anwenden eigener Markup-Sprachen* für deskriptives Markup in größerem Umfang eingesetzt (Abschluss der Entwicklung: 1986 als ISO-Standard 8879).

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Inhalt](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

## 3.1 Dokumenttypen

- In SGML wird jedem Dokument ein bestimmter *Dokumenttyp* zugeordnet (grundlegende Idee: bestimmte Gruppen von Dokumenten haben ähnliche Strukturen). Beispiele für solche Gruppen von Dokumenten sind Bücher, Briefe, Artikel, Berichte, usw.
- Die Struktur einer Klasse von Dokumenten wird in einer **Document Type Definition** (DTD) festgelegt.
- Welchen Typ ein konkretes Dokument besitzt, wird am Anfang des Dokumentes in der **Dokumenttypdeklaration** spezifiziert.
- Solche so vereinheitlichten Dokumente lassen sich leicht(er) weiter verarbeiten.

Ausgangsposition um  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

## 3.2 Deskriptives Markup

- SGML verwendet *deskriptives Markup*:
  - Ein SGML-Dokument enthält keine Angaben darüber, wie das Dokument konkret formatiert werden soll.
  - Das Dokument besteht nur aus den Textdaten und Informationen darüber, welche Textdaten zu welchen Textelementen gehören.
- Dies hat den Vorteil, dass ein einmal geschriebenes Dokument auf viele *unterschiedliche Arten formatiert* werden kann, ohne dass am eigentlichen Dokument etwas geändert werden muss.
- Zur Formatierung von SGML-Dokumenten könn(t)en spezielle Programme und DSSSL (*Document Style Semantics and Specification Language*, ISO 10179) eingesetzt werden.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Beis  
Informationsquellen (f  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

### 3.3 Aufbau eines SGML-Dokuments

Ein SGML-Dokument besteht aus drei Teilen:

- **Deklaration:** spezifiziert grundlegende Dinge, wie z. B. den verwendeten Zeichensatz, Länge der Bezeichner für Elemente, Festlegung der zugelassenen und verbotenen Zeichen, Regeln für gültige Bezeichner, Aktivierung bzw. Deaktivierung bestimmter Syntaxelemente, usw.
- **Dokument-Typ-Definition** (*Document Type Definition = DTD*): legt fest, welche Strukturelemente ein Dokument enthalten darf und wie sie ineinander geschachtelt sind)
- **Dokument-Instanz:** enthält die eigentlichen Textdaten zusammen mit dem dazugehörigen Markup und den Entity-Referenzen

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (I  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

## 3.4 HTML

Bekannteste SGML-Applikation ist **HTML** (*Hypertext Markup Language*) mit den drei Ausprägungen (hier für HTML 4.01):

- *HTML 4.01 Strict DTD*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html401/strict.dtd">
```

- *HTML 4.01 Transitional oder loose DTD*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html401/loose.dtd">
```

- *HTML 4.01 Frameset DTD*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html401/frameset.dtd">
```

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

# 4 XML

4.1	Gründe für XML	15
4.2	Entwurfsziele für XML	17
4.3	XML-Konzepte	18

---

Aus den Erfahrungen mit SGML und HTML wurde **XML** vom W3C als eine *stark vereinfachte SGML-Teilmenge* entwickelt:

- 1996: Beginn der Entwicklung
- 1997: *Proposed Recommendation*
- 1998: *Recommendation*
- 2000: *Extensible Markup Language (XML) 1.0 (Third Edition)*  
<http://www.w3.org/TR/xml>
- 2003: *Extensible Markup Language (XML) 1.1*  
<http://www.w3.org/TR/xml11/>

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

## 4.1 Gründe für XML

### Gewichtige Nachteile von HTML:

- Syntax und Bedeutung von Elementen und Attributen festgeschrieben;
- HTML eher auf die Präsentation des Textes am Bildschirm ausgerichtet;
- kaum strukturbeschreibende oder bedeutungsbeschreibende Elemente;
- keine klare Trennung zwischen Struktur, Inhalt und Darstellung;
- viele firmenspezifische Erweiterungen gegenüber dem Standard;
- viele Programme produzieren ungültigen HTML-Code;
- (ungültige) HTML-Dokumente, die zwar von einigen Browsern dargestellt werden, erschweren den Datenaustausch;
- nur eingeschränkte Unterstützung von Metadaten;
- viele strukturelle Möglichkeiten zum Datenaustausch fehlen;
- fehlende moderne Features, z. B. Dokument-Objekt-Modell

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (U  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**



## Gewichtige Nachteile von SGML:

- SGML ist hochkomplex und daher nur sehr schwer zu erlernen;
- SGML ist nur sehr aufwändig implementierbar;
- hoher Pflegeaufwand für SGML-Dokumente;
- SGML-Verarbeitungsprogramme sind meist hochspezialisiert und teuer;
- SGML-Dokumente sind i. allg. ungeeignet zur Darstellung im WWW.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 4.2 Entwurfsziele für XML

- XML-Dokumente sind *einfach zu erstellen*.
- XML-Dokumente sind *übersichtlich* und können ohne zusätzliche Hilfsmittel gelesen werden.
- Programme zum Verarbeiten von XML-Dateien sind *einfach zu entwickeln*.
- XML-Dokumente sind ähnlich wie HTML-Dokumente *schnell und einfach im WWW darstellbar*.
- XML ist *kompatibel mit SGML*.
- Minimierung des Markups ist nicht zulässig.
- XML enthält möglichst wenige optionale Features.
- XML ist für eine breite Anwendungspalette einsetzbar.
- **Der XML-Standard wird schnell verabschiedet.**

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (U  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

## 4.3 XML-Konzepte

4.3.1	„Wohlgeformte“ und gültige XML-Dateien	19
4.3.2	XML-Deklaration ( <i>XML declaration</i> )	21
4.3.3	Dokument-Typdeklaration ( <i>document type declaration</i> )	22
4.3.4	Element-Deklaration	23
4.3.5	Attribut-Deklaration	24
4.3.6	Entity-Deklaration	24

---

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

### 4.3.1 „Wohlgeformte“ und gültige XML-Dateien

Ein XML-Dokument, für das die folgenden Regeln gelten, ist „**wohlgeformt**“ (*well-formed*):

- Der Inhalt des Dokuments muss in genau einem *Wurzelement* eingeschlossen werden.
- Elemente mit optionalem *Ende-Tag* sind nicht zulässig.
- Alle *leeren Elemente* müssen geschlossen werden, also z. B. `<leer />` oder `<leer></leer>`.
- *Groß- und Kleinschreibung* bei Element- und Attribut-Namen wird unterschieden.
- *Überlappende Elemente* sind nicht erlaubt, also z. B. nicht  
`<wichtig>Hallo  
 <ganzwichtig>Welt</wichtig></ganzwichtig>`.
- Alle *Attribut-Werte* müssen in Anführungszeichen eingeschlossen werden.
- Es gibt nur noch einige *wenige vordefinierte Zeichen-Entities*:

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

<!ENTITY lt	"&#60;">	<!-- "<" -->
<!ENTITY gt	"&#62;">	<!-- ">" -->
<!ENTITY amp	"&#38;">	<!-- "&" -->
<!ENTITY apos	"&#39;">	<!-- "'>" -->
<!ENTITY quot	"&#34;">	<!-- '"'"' -->

Ein wohlgeformtes XML-Dokument, das einer DTD gehorcht, ist *gültig* (**valid**).

Ausgangsposition un  
 Markup  
 SGML  
 XML  
 XML – Begleitende  
 Standardisierte XML  
 Arbeitsabläufe – Übe  
 Arbeitsabläufe – Beis  
 Informationsquellen  
 URLs  
 Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

### 4.3.2 XML-Deklaration (*XML declaration*)

In der **XML-Deklaration** wird – falls vorhanden – die XML-Version und optional der zugrunde liegende Zeichensatz spezifiziert:

```
<?xml version="1.0" encoding='ISO-8859-1' ?>
```

oder

```
<?xml version="1.0" standalone="yes" ?>
```

Mit der Angabe `standalone="yes"` teilt man dem XML-Prozessor mit, dass er nicht nach externen Bestandteilen (z. B. nach einer DTD) suchen soll. Voreingestellter Zeichensatz ist übrigens **UTF-8** bzw. **UTF-16**.

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Inhalt](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

### 4.3.3 Dokument-Typdeklaration (*document type declaration*)

Die DTD, die einem XML-Dokument zugrunde liegt, kann am Anfang des Dokuments spezifiziert werden:

1.

```
<?xml version="1.0" ?>
<!DOCTYPE adressen SYSTEM "adressen.dtd">
...
```

bzw.

2.

```
<?xml version="1.0" ?>
<!DOCTYPE adressen [
  <!ELEMENT adressen ...>
  <!ENTITY ...>
  ...
  <!ATTLIST ...>
]>
```

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

### 4.3.4 Element-Deklaration

Alle Elemente werden nach dem folgenden allgemeinen Schema vereinbart:

```
<!ELEMENT name inhalt>
```

mit EMPTY, ANY, *inhaltsmodell* als Möglichkeiten für *inhalt*.

- Im *inhaltsmodell* werden der Inhaltstyp, die möglichen Tochter-Elemente, ihre Reihenfolge bzw. ihre Häufigkeit aufgeführt. Beispiele:

```
<!ELEMENT sammlung (gedicht*)           >
<!ELEMENT gedicht  (kopf, strophe+)      >
<!ELEMENT kopf     (titel?,autor?,jahr?) >
<!ELEMENT titel    (#PCDATA)             >
```

...

- *sammlung* enthält beliebig viele (\*) Elemente der Art *gedicht*.
- Ein *gedicht* besteht aus genau einem *kopf* und mindestens einer (+) *strophe*.
- Der *kopf* besteht aus den optionalen (?) Teilen *titel*, *autor*, *jahr* (in dieser Reihenfolge).
- Das Element *titel* enthält Daten, die weiter verarbeitet werden können (#PCDATA).

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)



### 4.3.5 Attribut-Deklaration

XML-Elemente können nach folgendem Schema zusätzlich durch Attribute genauer festgelegt werden:

```
<!ATTLIST element name wertebereich voreinstellung
...>
```

- Für *wertebereich* wird am häufigsten CDATA (Zeichenkette) oder eine *aufzählung* angegeben.
- Üblich für *voreinstellung* sind #REQUIRED (zwingend), #IMPLIED (implizit) oder ein konkreter Weglasswert.

#### Beispiel:

```
<!ATTLIST bild name      CDATA #REQUIRED
               groesse   CDATA #IMPLIED
               titel      CDATA "Weglasstitel"
               typ        (inline|display) 'inline'>
```

### 4.3.6 Entity-Deklaration

Mit Hilfe von *Entities* kann zusätzliches Material (Textfragmente, Spezialzeichen, Bilder, externe Dateien) in ein XML-Dokument eingebracht werden.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

## 5 XML – Begleitende Konzepte (Auswahl)

- 29.06.1999: *Associating Style Sheets with XML documents*
- 15.03.2001: *Canonical XML Version 1.0*
- 11.01.1999: *Cascading Style Sheets (CSS1) Level 1 Specification*
- 12.05.1998: *Cascading Style Sheets, level 2 (CSS2) Specification*
- 10.12.2002: *Decryption Transform for XML Signature*
- 01.10.1998: *Document Object Model (DOM) Level 1*
- 13.11.2000: *Document Object Model (DOM) Level 2 Core Specification*
- 13.11.2000: *Document Object Model (DOM) Level 2 Events Specification*
- 07.01.2003: *Document Object Model (DOM) Level 2 HTML Specification*

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende Konzepte  
Standardisierte XML  
Arbeitsabläufe – Überblick  
Arbeitsabläufe – Beispiel  
Informationsquellen (Literatur)  
URLs  
Nachtrag: XSL Formate

**Suchen**

**Abbruch**

- 13.11.2000: *Document Object Model (DOM) Level 2 Style Specification*
- 13.11.2000: *Document Object Model (DOM) Level 2 Traversal and Range Specification*
- 13.11.2000: *Document Object Model (DOM) Level 2 Views Specification*
- 27.01.2004: *Document Object Model (DOM) Level 3 Validation Specification*
- 18.07.2002: *Exclusive XML Canonicalization Version 1.0*
- 04.02.2004: *Extensible Markup Language (XML) 1.0 (Third Edition)*
- 04.02.2004: *Extensible Markup Language (XML) 1.1*
- 15.10.2001: *Extensible Stylesheet Language (XSL) Version 1.0*
- 07.07.1999: *Mathematical Markup Language (MathML) 1.01 Specification*
- 21.10.2003: *Mathematical Markup Language (MathML) Version 2.0 (Second Edition)*

Ausgangsposition un  
 Markup  
 SGML  
 XML  
 XML – Begleitende  
 Standardisierte XML  
 Arbeitsabläufe – Übe  
 Arbeitsabläufe – Beis  
 Informationsquellen  
 URLs  
 Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

- 10.04.2001: *Modularization of XHTML(tm)*
- 14.01.1999: *Namespaces in XML*
- 04.02.2004: *Namespaces in XML 1.1*
- 14.10.2003: *XForms 1.0*
- 01.08.2002: *XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)*
- 31.05.2001: *XHTML 1.1 - Module-based XHTML*
- 19.12.2000: *XHTML Basic*
- 27.06.2001: *XML Base*
- 10.12.2002: *XML Encryption Syntax and Processing*
- 14.10.2003: *XML Events*
- 04.02.2004: *XML Information Set (Second Edition)*
- 27.06.2001: *XML Linking Language (XLink) Version 1.0*
- 16.11.1999: *XML Path Language (XPath) Version 1.0*

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

- 02.05.2001: *XML Schema Part 0: Primer*
- 02.05.2001: *XML Schema Part 1: Structures*
- 02.05.2001: *XML Schema Part 2: Datatypes*
- 12.02.2002: *XML-Signature Syntax and Processing*
- 08.11.2002: *XML-Signature XPath Filter 2.0*
- 25.03.2003: *XPointer element() Scheme*
- 25.03.2003: *XPointer Framework*
- 25.03.2003: *XPointer xmlns() Scheme*
- 16.11.1999: *XSL Transformations (XSLT) Version 1.0*

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

# 6 Standardisierte XML-Applikationen

6.1	XHTML	31
6.2	MathML	35

---

Eine Auswahl aus den schon standardisierten XML-Anwendungen:

- *DocBook*
- *Text Encoding Initiative*
- *XHTML 1.0: The Extensible HyperText Markup Language – A Reformulation of HTML 4 in XML 1.0*
- MathML (Mathematical Markup Language): *Mathematical Markup Language (MathML) 1.01 Specification* und *W3C's Math Home Page*;

Ausführliche Dokumentation bei Hans Hagen:

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (f  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

- *MathML* (Handbuch),
  - *MATHML* (Übersicht) und
  - *MathML in ConT<sub>E</sub>Xt* (Beispiele)
- und viele andere

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 6.1 XHTML

Ursprüngliche HTML-Datei: xhtml-a.html

```
1  <html lang=de>
2  <Meta http-equiv=Content-Type
3      content="text/html; charset=ISO-8859-1">
4  <meta name=ROBOTS content="INDEX, FOLLOW">
5  <META name=KEYWORDS lang=de content="Vortrag, Test">
6  <meta name=DESCRIPTION lang=de
7      content="Vortrag für MV27">
8  <meta name=Author
9      content="Guenter.Partosch@hrz.uni-giessen.de">
10 <head><title>Ein kleiner Versuch</title></Head>
11 <body>
12 <h1>Überschrift</h1>
13
14 <p>Ein Absatz<br>
15 Mit einer neuen Zeile
16 <p>neuer Absatz
17 <ul>
18 <li>erster Eintrag
19 <li>zweiter Eintrag
```

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch



```
19 </ul>
20 <hr>
21 Ende
22 </BODY>
23 </HTML>
```

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Beis  
Informationsquellen (I  
URLs  
Nachtrag: XSL Form

Bei der Umsetzung durch **Tidy** erhält man beispielsweise  
[Tidy: *Clean up your Web pages with HTML TIDY*  
(<http://www.w3.org/People/Raggett/tidy/>)]:

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

Suchen

Abbruch

Dazu passende XHTML-Datei: xhtml-b.html

```
1  <?xml encoding="iso-8859-1" version="1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml" lang="de">
5  <head>
6  <meta content="HTML Tidy for Windows (vers 1st November 2002),
7      see www.w3.org" name="generator" />
8  <meta http-equiv="Content-Type"
9      content="text/html; charset=ISO-8859-1" />
10 <meta name="ROBOTS" content="INDEX, FOLLOW" />
11 <meta name="KEYWORDS" lang="de" content="Vortrag, Test" />
12 <meta name="DESCRIPTION" lang="de"
13     content="Vortrag für MV27" />
14 <meta name="Author"
15     content="Guenter.Partosch@hrz.uni-giessen.de" />
16 <title>Ein kleiner Versuch</title>
17 </head>
18 <body>
```

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

```
19 <h1>Überschrift</h1>
20 <p>Ein Absatz<br />
21 Mit einer neuen Zeile</p>
22 <p>neuer Absatz</p>
23 <ul>
24 <li>erster Eintrag</li>
25 <li>zweiter Eintrag</li>
26 </ul>
27 <hr />
28 Ende
29 </body>
30 </html>
```

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Beis  
Informationsquellen (I  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 6.2 MathML

**MathML (präsentationsorientiert):** mathml-a.xml

**Ergebnis**

```
1  <math>
2    <mrow>
3      <mrow>
4        <msup><mi>x</mi><mn>2</mn></msup>
5        <mo>+</mo>
6        <mrow>
7          <mn>4</mn>
8          <mo>&InvisibleTimes;</mo>
9          <mi>x</mi>
10       </mrow>
11     <mo>+</mo><mn>4</mn>
12   </mrow>
13   <mo>=</mo><mn>0</mn>
14 </mrow>
15 </math>
```

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

```
1 <math>
2   <apply><eq />
3     <apply><power /> <ci>a</ci> <ci>2</ci> </apply>
4     <apply><plus />
5       <apply><power /> <ci>b</ci> <ci>2</ci> </apply>
6       <apply><power /> <ci>c</ci> <ci>2</ci> </apply>
7     </apply>
8   </apply>
9 </math>
```

Ausgangsposition um  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

# 7 Arbeitsabläufe – Überblick

7.1	XML-Werkzeuge und -Software	38
7.2	Erstellen von XML-Dateien	39
7.3	Darstellen von XML-Dateien	40
7.4	Konvertieren von XML-Dateien	46
7.5	Resümee	56

---

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 7.1 XML-Werkzeuge und -Software

Für alles, was

- Erstellen,
- Konvertieren,
- Bearbeiten und
- Darstellen

von XML-Dateien betrifft, ist

*Free XML tools and software*

(<http://www.garshol.priv.no/download/xmltools/>)

eine gute Quelle.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (d  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

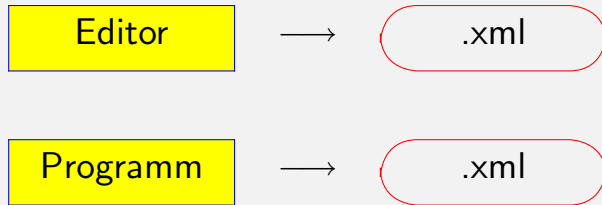
**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 7.2 Erstellen von XML-Dateien



Ausgangsposition um  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**



## 7.3 Darstellen von XML-Dateien

7.3.1	Direkte Darstellung von XML-Dateien	41
7.3.2	Darstellen von PDF-Dateien	44
7.3.3	Anmerkungen zu PDF	45

---

*Darstellen von XML-Dokumenten* meint (fast) immer

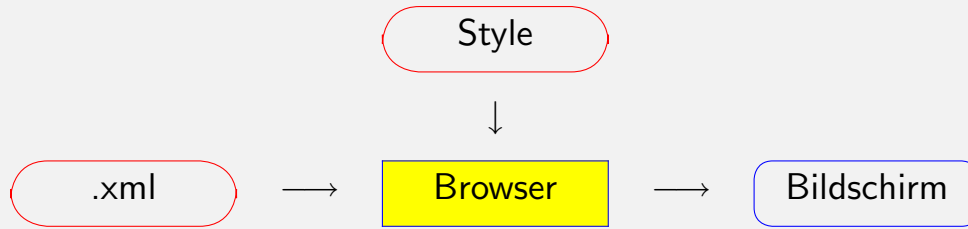
- **Präsentieren am Bildschirm**, z. B. mit Hilfe eines Browsers; ggf. vorher Umwandlung in HTML
- **PDF-Dokument** ausdrucken bzw. am Bildschirm präsentieren  $\Rightarrow$  vorher: PDF-Datei erstellen

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

## 7.3.1 Direkte Darstellung von XML-Dateien



Bei den Style-Dateien handelt es sich je nach Browser/Version/Anwendung um

- .xsl-Dateien oder
- .css-Dateien.

Ausgangsposition um  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## Direkte Darstellung mittels CSS

Mit Hilfe von *Cascading Style Sheets* (*Cascading Style Sheets (CSS1) Level 1 Specification*, *Cascading Style Sheets, level 2 (CSS2) Specification*) lässt sich eine XML-Datei relativ einfach in einem XML-fähigen Browser darstellen:

### Nachteile:

- Auswertung der Attribut-Werte nicht generell möglich (insbesondere nicht in CSS1)
- Ausgabe zusätzlicher Texte nur eingeschränkt möglich
- Änderung der vorgegebenen Reihenfolge nicht möglich
- Wiederverwendung von Elementinhalten nicht möglich
- CSS (noch immer) uneinheitlich bzw. unvollständig in den Browsern realisiert

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Beis  
Informationsquellen (f  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

## Direkte Darstellung mittels XSL (*Extensible Style Language*)

Mit Hilfe einer XSL-Style-Datei (*Extensible Stylesheet Language (XSL) Version 1.0* und *XSL Transformations (XSLT) Version 1.0*) lässt sich eine XML-Datei relativ einfach in einem XML-fähigen Browser umformen bzw. darstellen:

### Vorteile:

- sehr mächtige, XML-artige Programmiersprache
- (Fast) alle gewünschten Umformungen und Formatierungen lassen sich bewerkstelligen.
- leicht zu erlernen
- Alle oben genannte Nachteile von CSS sind behoben.

### Nachteile:

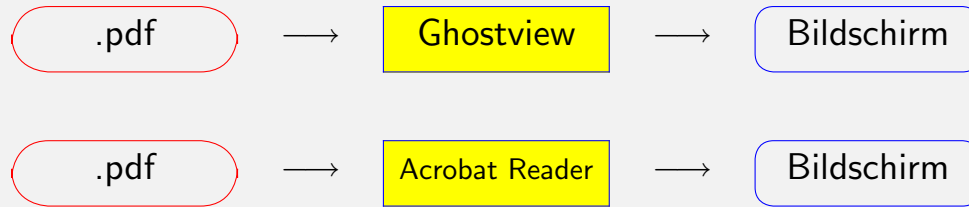
- existiert erst seit dem 15.10.2001 als vollständiger Standard (vorher lediglich der Teil, der sich mit Transformationen beschäftigt)
- Nicht alle Browser können XSL/XSLT auswerten.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

## 7.3.2 Darstellen von PDF-Dateien



Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

### 7.3.3 Anmerkungen zu PDF

- **Portable Document Format** (PDF) ist ein portables Darstellungsformat für Dokumente.
- Es ist weltweiter, geräte- und herstellerunabhängiger Standard.
- PDF wurde entwickelt von der Fa. Adobe; aktuelle Version ist 1.4 (Acrobat 5).
- Es basiert auf PostScript; verzichtet aber auf die Programmierbarkeit und enthält dafür ausgefeilte Möglichkeiten für Hypertextstrukturen.
- PDF beinhaltet einen gut durchdachten Font-Ersatz-Mechanismus (multiple master fonts).
- PDF ist ein Dokument-Endformat, d. h. nur noch wenige, spezielle Änderungen am Dokument sind möglich.
- PDF-Dokumente lassen sich gegen Drucken, Verändern, Entnehmen schützen.
- PDF-Dokumente können überall mit geeignetem Viewer betrachtet bzw. auf Drucker ausgedruckt werden.
- PDF-Dateien sind deutlich kleiner als entsprechende PostScript-Dateien.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

## 7.4 Konvertieren von XML-Dateien

7.4.1	Direkte Konvertierung	47
7.4.2	Konvertierung nach HTML bzw. Zwischenformaten	48
7.4.3	Konvertierung der Zwischenformate	51
7.4.4	Tool-Ketten	52
7.4.5	Warum sollte hier T <sub>E</sub> X eine Rolle spielen?	55

---

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

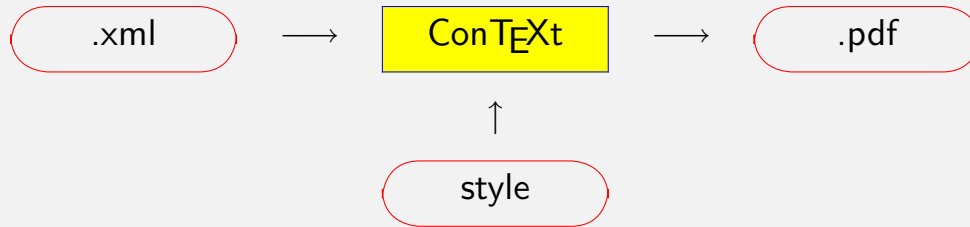
**Erste**

**Vorherige**

**Nächste**

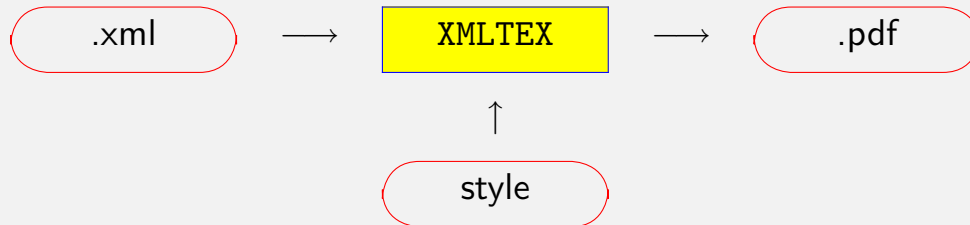
**Letzte**

## 7.4.1 Direkte Konvertierung



```
texexec --pdf -env=tut7.dir adr6.xml  
⇒ adr6.pdf
```

**Anmerkung:** ConTeXt wird hier mit texexec aufgerufen.



```
pdfxmltex adr6.xml
```

mit Direktivendatei `adressen.xmt` ⇒ `adr6.pdf`

**Anmerkung:** XMLTEX wird hier mit pdfxmltex aufgerufen.

Ausgangsposition um  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

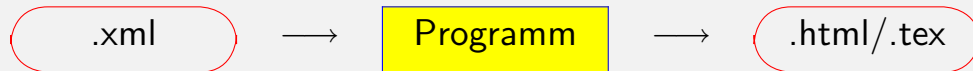
[Abbruch](#)



## 7.4.2 Konvertierung nach HTML bzw. Zwischenformaten

### Umsetzung mit Hilfe eigener, sehr spezieller Programme:

- Da gültige XML-Dokumente auch gültige SGML-Dokumente sind, lassen sich eigentlich alle SGML-Transformations-Tools einsetzen.
- Da Syntax und Struktur bei XML einfach sind, lassen sich mit geringem Aufwand auch eigene Programme für diesen (einen) Zweck entwickeln:



```
gawk -f zerteil.awk kapitel2.xml |  
gawk -f kap2t.awk > kapitel2.tex
```

oder

```
gawk -f zerteil.awk kapitel2.xml |  
gawk -f kap2h.awk > kapitel2.html
```

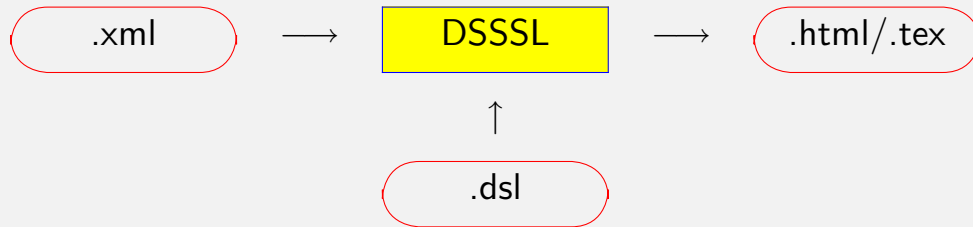
**Anmerkung:** gawk ist hier der Aufruf des Programms AWK.

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

## Umsetzung mit Hilfe von DSSSL (Document Style Semantics and Specification Language) und DSSSL-Prozessoren:



### Vorteile von DSSSL:

sehr mächtig; Fast alle gewünschten Umformungen und Formatierungen lassen sich bewerkstelligen.

### Nachteile von DSSSL:

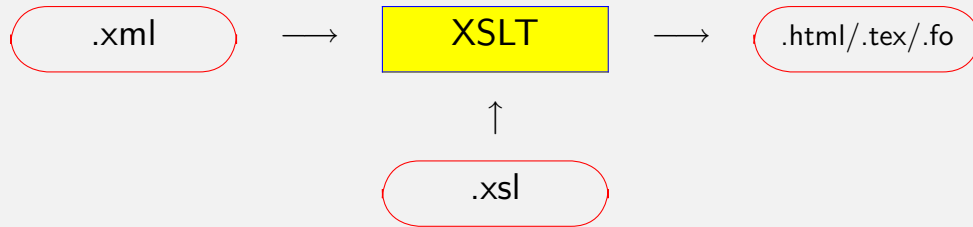
viel zu komplex und aufwändig viel zu schwierig zu erlernen (LISP-artige Programmierung!)

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

## Umsetzung mit Hilfe von XSLT-Prozessoren:



saxon kapitel2.xml kapitel2-tex1.xsl > kapitel2a.tex

oder

saxon kapitel2.xml kapitel2-html1.xsl > kapitel2.html

### Anmerkungen:

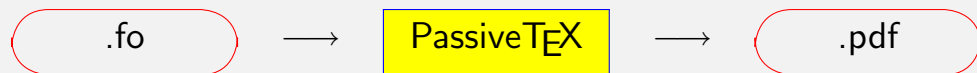
- .fo steht für *Formatting Objects*: sehr detaillierte Beschreibung des Dokumenten-Layouts im XML-Stil
- **saxon**: sehr leistungsfähiger und gut dokumentierter XSLT-Prozessor

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

## 7.4.3 Konvertierung der Zwischenformate



`pdfxmltex meine.xml`

---



**FOP** aus dem Apache-Projekt ist ein geeigneter Prozessor.

---



`pdftex meine.tex`

An Stelle von pdfTeX kann ggf. auch ConTeXt oder pdf<sup>L</sup>TeX benutzt werden.

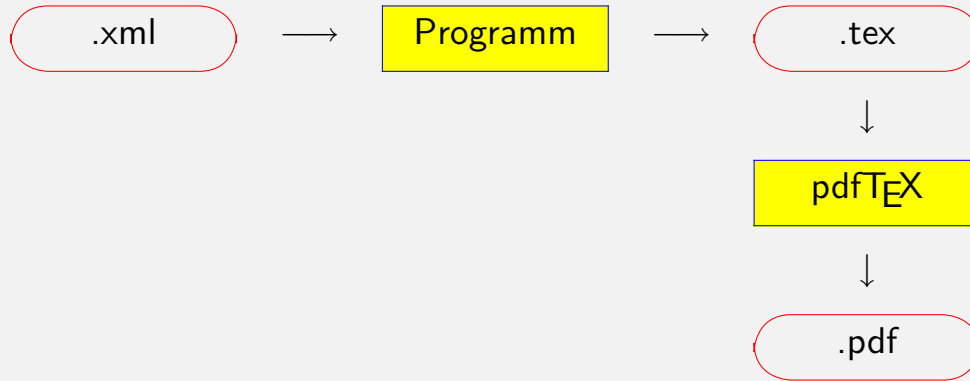
Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

## 7.4.4 Tool-Ketten

$XML \Rightarrow [Programm] \Rightarrow [pdfT_EX] \Rightarrow PDF$



```
gawk -f zerteil.awk kapitel2.xml |  
    gawk -f kap2t.awk > kapitel2.tex  
pdflatex kapitel2.tex  
 $\Rightarrow$  kapitel2.pdf
```

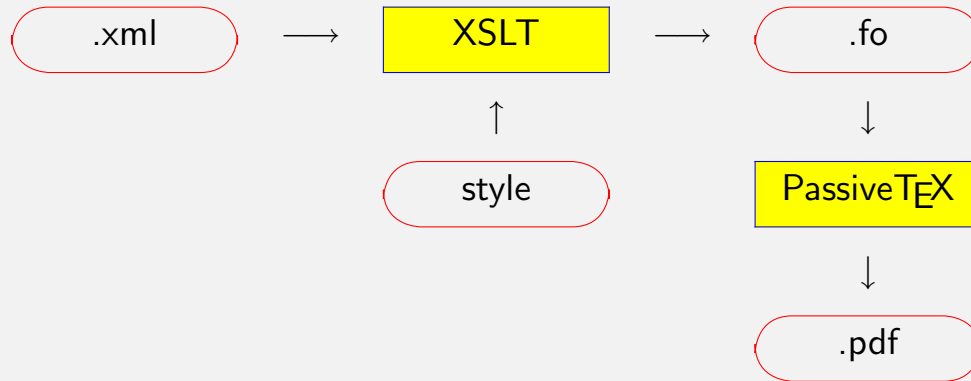
**Anmerkung:** hier konkret pdf<sup>L</sup>T<sub>E</sub>X für pdfT<sub>E</sub>X eingesetzt

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

$XML \Rightarrow [XSLT] \Rightarrow [PassiveT_E X] \Rightarrow PDF$



```
saxon meine.xml meine.xsl > meine-fo.xml  
pdfxmltex meine-fo.xml
```

**Anmerkung:** hier konkret pdfxmltex für PassiveT<sub>E</sub>X eingesetzt

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

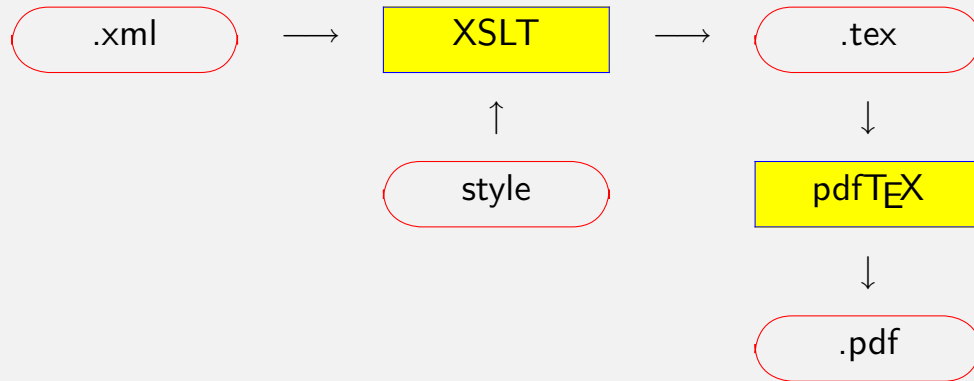
Erste

Vorherige

Nächste

Letzte

$XML \Rightarrow [XSLT] \Rightarrow [pdfT_E X] \Rightarrow PDF$



```
saxon kapitel2.xml kapitel2-tex1.xsl > kapitel2a.tex
pdflatex kapitel2a.tex
⇒ kapitel2a.pdf
```

**Anmerkung:** hier konkret pdf<sup>L</sup>T<sub>E</sub>X für pdfT<sub>E</sub>X eingesetzt

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

## 7.4.5 Warum sollte hier T<sub>E</sub>X eine Rolle spielen?

- Dokumente, die hohen *typographischen Anforderungen* genügen, werden immer benötigt. – Und T<sub>E</sub>X erfüllt diese Qualitätsansprüche.
- T<sub>E</sub>X kann *qualitativ hochwertige PDF-Dateien* erzeugen.
- T<sub>E</sub>X ist *fehlerfrei* und läuft *nahezu stabil*.
- T<sub>E</sub>X läuft auf Wunsch *batch-artig* ab und kann hervorragend in Tool-Ketten eingesetzt werden.
- T<sub>E</sub>X ist *schnell* und kann auch sehr große Dokumente aufbereiten.
- Die *Darstellung mathematischer Formeln* in T<sub>E</sub>X ist unübertroffen gut.
- Trennungen durch T<sub>E</sub>X sind gut.
- **Wir kennen T<sub>E</sub>X (plainT<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, ConT<sub>E</sub>Xt, usw.)!**

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (u  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch



## 7.5 Resümee

.xml	⇒	Programm	⇒	.tex	⇒	pdfT <sub>E</sub> X	⇒	.pdf
.xml	⇒	XSLT	⇒	.tex	⇒	pdfT <sub>E</sub> X	⇒	.pdf
.xml	⇒				⇒	XMLT <sub>E</sub> X	⇒	.pdf
.xml	⇒	XSLT	⇒	.fo	⇒	PassiveT <sub>E</sub> X	⇒	.pdf
.xml	⇒	XSLT	⇒	.fo	⇒	FOP	⇒	.pdf
.xml	⇒				⇒	ConT <sub>E</sub> Xt	⇒	.pdf

Ausgangsposition um  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

# 8 Arbeitsabläufe – Beispiele

8.1	Gedichtsammlung	58
8.2	Tabelle	67

---

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 8.1 Gedichtsammlung

Eine Gedichtsammlung, die z.Zt. nur aus einem Gedicht besteht, wird *wohlgeformt* mit XML beschrieben.

8.1.1	Erstellen der XML-Datei	59
8.1.2	Darstellen der XML-Datei	64

---

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 8.1.1 Erstellen der XML-Datei

Eine XML-Datei lässt sich mit einem einfachen Texteditor erstellen und bearbeiten:

```
1  <sammlung>
2    <gedicht>
3      <kopf>
4        <titel>The SICK ROSE</titel>
5        <autor>Unknown Poet</autor>
6        <jahr/>
7      </kopf>
8      <strophe>
9        <zeile>O Rose thou art sick.</zeile>
10       <zeile>The invisible worm,</zeile>
11       <zeile>That flies in the night</zeile>
12       <zeile>In the howling storm:</zeile>
13     </strophe>
14     <strophe>
15       <zeile>Has found out thy bed</zeile>
16       <zeile>Of crimson joy:</zeile>
17       <zeile>And his dark secret love</zeile>
18       <zeile>Does thy life destroy.</zeile>
19     </strophe>
```

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (I  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

```
20      <strophe>
21          <zeile>Has found out thy bed</zeile>
22          <zeile>Of crimson joy:</zeile>
23          <zeile>And his dark secret love</zeile>
24          <zeile>Does thy life destroy.</zeile>
25      </strophe>
26  </gedicht>
27 </sammlung>
```

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

Dieser Darstellung liegt eine vorher entwickelte *DTD* zugrunde:

```
1 <!ELEMENT sammlung (gedicht*) >
2 <!ELEMENT gedicht (kopf?, strophe+) >
3 <!ELEMENT kopf (titel?,autor?,jahr?) >
4 <!ELEMENT titel (#PCDATA) >
5 <!ELEMENT autor (#PCDATA) >
6 <!ELEMENT jahr (#PCDATA) >
7 <!ELEMENT strophe (zeile+) >
8 <!ELEMENT zeile (#PCDATA) >
```

**Anmerkung:** Man könnte die DTD sich auch erzeugen lassen:

*DTDGenerator – A tool to generate XML DTDs*

<http://saxon.sourceforge.net/dtdgen.html>

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Beide Teile lassen sich zu der vollständigen XML-Datei gedicht0.xml zusammenfassen:

```
1  <?xml version="1.0"?>
2  <!-- gedicht0.xml -->
3  <!DOCTYPE sammlung [
4  <!ELEMENT sammlung (gedicht*) >
5  <!ELEMENT gedicht (kopf?, strophe+) >
6  <!ELEMENT kopf (titel?, autor?, jahr?) >
7  <!ELEMENT titel (#PCDATA) >
8  <!ELEMENT autor (#PCDATA) >
9  <!ELEMENT jahr (#PCDATA) >
10 <!ELEMENT strophe (zeile+) >
11 <!ELEMENT zeile (#PCDATA) >
12 ]>
13 <sammlung>
14     <gedicht>
15         <kopf>
16             <titel>The SICK ROSE</titel>
17             <autor>Unknown Poet</autor>
18             <jahr/>
19         </kopf>
20         <strophe>
```

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

```

21      <zeile>O Rose thou art sick.</zeile>
22      <zeile>The invisible worm,</zeile>
23      <zeile>That flies in the night</zeile>
24      <zeile>In the howling storm:</zeile>
25  </strophe>
26  <strophe>
27      <zeile>Has found out thy bed</zeile>
28      <zeile>Of crimson joy:</zeile>
29      <zeile>And his dark secret love</zeile>
30      <zeile>Does thy life destroy.</zeile>
31  </strophe>
32  <strophe>
33      <zeile>Has found out thy bed</zeile>
34      <zeile>Of crimson joy:</zeile>
35      <zeile>And his dark secret love</zeile>
36      <zeile>Does thy life destroy.</zeile>
37  </strophe>
38  </gedicht>
39 </sammlung>

```

Ausgangsposition un  
 Markup  
 SGML  
 XML  
 XML – Begleitende  
 Standardisierte XML  
 Arbeitsabläufe – Übe  
 Arbeitsabläufe – Beis  
 Informationsquellen  
 URLs  
 Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte



## 8.1.2 Darstellen der XML-Datei

Auf nicht-XML-fähigen Browsern (z.B. Netscape 4.7, Internet Explorer 4) lässt sich diese XML-Datei nur unzureichend darstellen. Im Internet Explorer 5ff wird wenigstens die Struktur der XML-Datei dargestellt: **Ergebnis**

Entwickelt man für diese XML-Datei die dazu passende CSS-Datei `gedicht.css`:

```
1  /* gedicht.css */
2  sammlung {color:#CC0000;
3           font-size:110%; }
4  gedicht  {display:block;
5           margin-top:2.0ex; }
6  titel    {display:block;
7           font-weight:bold;
8           margin-bottom:0.5ex; }
9  autor    {display:block;
10          font-size: 80%; }
11 strophe  {display:block;
12          margin-top:1.0ex;
13          margin-left:0.5cm; }
14 zeile    {display:block; }
```

und ergänzt die XML-Datei entsprechend:

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (I  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

```

1  <?xml version="1.0"?>
2  <!-- gedicht.xml -->
3  <?xml-stylesheet href="gedicht.css" type="text/css" ?>
4  <!DOCTYPE sammlung [
5  <!ELEMENT sammlung (gedicht*) >
6  <!ELEMENT gedicht (kopf?, strophe+) >
7  <!ELEMENT kopf (titel?,autor?,jahr?) >
8  <!ELEMENT titel (#PCDATA) >
9  <!ELEMENT autor (#PCDATA) >
10 <!ELEMENT jahr (#PCDATA) >
11 <!ELEMENT strophe (zeile+) >
12 <!ELEMENT zeile (#PCDATA) >
13 ]>
14 <sammlung>
15     <gedicht>
16         <kopf>
17             <titel>The SICK ROSE</titel>
18             <autor>Unknown Poet</autor>
19             <jahr/>
20         </kopf>
21         <strophe>
22             <zeile>O Rose thou art sick.</zeile>
23             <zeile>The invisible worm,</zeile>

```

Ausgangsposition un  
 Markup  
 SGML  
 XML  
 XML – Begleitende  
 Standardisierte XML  
 Arbeitsabläufe – Übe  
 Arbeitsabläufe – Beis  
 Informationsquellen  
 URLs  
 Nachtrag: XSL Form

Suchen

Abbruch

```
24      <zeile>That flies in the night</zeile>
25      <zeile>In the howling storm:</zeile>
26  </strophe>
27  <strophe>
28      <zeile>Has found out thy bed</zeile>
29      <zeile>Of crimson joy:</zeile>
30      <zeile>And his dark secret love</zeile>
31      <zeile>Does thy life destroy.</zeile>
32  </strophe>
33  <strophe>
34      <zeile>Has found out thy bed</zeile>
35      <zeile>Of crimson joy:</zeile>
36      <zeile>And his dark secret love</zeile>
37      <zeile>Does thy life destroy.</zeile>
38  </strophe>
39  </gedicht>
40 </sammlung>
```

so erhält man in XML-fähigen Browsern beispielsweise die folgende Darstellung: **Ergebnis**

- Ausgangsposition und
- Markup
- SGML
- XML
- XML – Begleitende
- Standardisierte XML
- Arbeitsabläufe – Üb
- Arbeitsabläufe – Beis
- Informationsquellen
- URLs
- Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

## 8.2 Tabelle

8.2.1	Erstellen der XML-Datei	68
8.2.2	Darstellen der XML-Datei	68
8.2.3	Umformen durch spezialisiertes Programm	68
8.2.4	Darstellen mittels CSS-Datei und Browser	69
8.2.5	Transformation in eine HTML-Datei mittels XSL(T) und Browser	69
8.2.6	Transformation in eine HTML-Datei (mit Sortierung)	70
8.2.7	Transformation der XML-Datei in eine L <sup>A</sup> T <sub>E</sub> X-Datei	71

---

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 8.2.1 Erstellen der XML-Datei

Ausgangspunkt könnte eine WinWord-Tabelle sein: **Zeigen**  
oder eine daraus abgeleitete ASCII-Rohfassung (`adr.txt`): **Zeigen**

Diese Datei lässt sich mit geringem Aufwand mittels eines spezialisierten AWK-Programms (`xml-adr.awk`) in eine XML-Datei (`adr.xml`) umwandeln:

```
gawk -f xml-adr.awk adr.txt > adr.xml
```

oder alternativ als XML-Datei `adr2.xml` mit externer DTD `adressen.dtd`.

## 8.2.2 Darstellen der XML-Datei

Diese XML-Datei `adr2.xml` lässt sich zwar nicht im Internet Explorer 4 oder Netscape 4.7 darstellen, wohl aber ihre Struktur ohne weiteren Aufwand im Internet Explorer 5ff oder XML Notepad: **Zeigen**

## 8.2.3 Umformen durch spezialisiertes Programm

Die XML-Datei `adr2.xml` kann durch ein spezialisiertes Programm, wie z.B. AWK in eine HTML-Datei umgeformt werden:

```
gawk -f zerteil.awk adr2.xml | gawk -f pr1.awk > adr.html
```

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

## 8.2.4 Darstellen mittels CSS-Datei und Browser

Entwickelt man für die XML-Datei `adr2.xml` eine spezielle CSS-Datei (`adr.css`) und ändert die XML-Datei entsprechend ab (`adr3.xml`), so erhält man bei Mozilla und Netscape 6 die folgende Darstellung: **Zeigen**

## 8.2.5 Transformation in eine HTML-Datei mittels XSL(T) und Browser

Weitaus mehr Möglichkeiten bietet die Verbindung der XML-Datei (`adr5.xml`) mit einer speziellen XSL(T)-Datei (`adr.xsl`): **Ergebnis**

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Inhalt](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

## 8.2.6 Transformation in eine HTML-Datei (mit Sortierung)

Mit Hilfe der XSL-Datei `adr2.xsl` lässt sich die XML-Datei `adr6.xml` in einer sortierten Tabelle ausgeben: **Ergebnis**

### Vorsicht:

Die Zeile (25) in `adr2.xsl`

```
<xsl:for-each select="adressen/adresse" order-by="ort/plz">
```

entstammt einem alten XSL-Standard und ist heute nicht mehr gültig. Sie müsste jetzt wie folgt heißen:

```
<xsl:for-each select="adressen/adresse">  
  <xsl:sort select="ort/plz" />
```

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Inhalt](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

## 8.2.7 Transformation der XML-Datei in eine L<sup>A</sup>T<sub>E</sub>X-Datei

Mit Hilfe des XSLT-Prozessors **saxon** wird das XML-Dokument `adr5.xml` in die L<sup>A</sup>T<sub>E</sub>X-Datei `adr-tex.tex` transferiert. Gesteuert wird dabei der Vorgang durch die XSL-Style-Datei `adr-tex.xsl`.

Der Aufruf

```
saxon adr5.xml adr-tex.xsl > adr-tex.tex
```

führt zur L<sup>A</sup>T<sub>E</sub>X-Datei `adr-tex.tex`.

Diese L<sup>A</sup>T<sub>E</sub>X-Datei wird mittels des Aufrufs

```
latex adr-tex.tex
```

weiter verarbeitet; die Ergebnisdatei `adr5.dvi` kann durch einen DVI-Previewer am Bildschirm betrachtet werden: **Ergebnis**.

Die konkrete Formatierung wird dabei durch das L<sup>A</sup>T<sub>E</sub>X-Package `adr-tex.sty` festgelegt.

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)



# 9 Informationsquellen (Auswahl)

9.1	Usenet-News-Gruppen und Diskussionslisten	73
9.2	Anleitungen im Netz	74
9.3	Bücher (Auswahl)	75

---

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 9.1 Usenet-News-Gruppen und Diskussionslisten

- `comp.text.xml` (Usenet-Gruppe; englisch)
  - `microsoft.public.xml` (Usenet-Gruppe; englisch)
  - `microsoft.public.xsl` (Usenet-Gruppe; englisch)
  - `de.comp.text.xml` (Usenet-Gruppe; deutsch)
  - `microsoft.de.german.public.xml` (Usenet-Gruppe; deutsch)
- 

- `xml-de@listserv.dfn.de` (E-Mail-Diskussionsliste, deutsch)
- `xml-l@LISTSERV.HEANET.IE` (E-Mail-Diskussionsliste, deutsch)

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

## 9.2 Anleitungen im Netz

- Microsoft: *XML Tutorial*;  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk30/htm/xmtutxmltutorial.asp>
- Microsoft: *XSLT Developer's Guide*;  
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/xmlsdk30/htm/xmconxsltdevelopersguide.asp>

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

## 9.3 Bücher (Auswahl)

- **Bach, Mike:** *XSL und XPath - verständlich und praxisnah. Transformation und Ausgabe von XML-Dokumenten mit XSL*; Addison-Wesley, München; 2001; ISBN 3-8273-16661-8
- **Behme, Henning; Mintert, Stefan:** *XML in der Praxis. Professionelles Web-Publishing mit der Extensible Markup Language*; Addison-Wesley, München; 2000; ISBN 3-82731-636-7
- **Harold, Elliotte Rusty; Means, W. Scott:** *XML in a Nutshell*; O'Reilly, Köln; 2003; ISBN 3-89721-337-0
- **Holzner, Steven:** *XML Insider. Das Expertenwissen zum De-Facto-Dokumentenstandard. XML auf der Schnittstelle zu den aktuellen Web-technologien*; Markt+Technik, München; 2001; ISBN 3-8272-6090-6
- **Kay, Michael:** *XSLT – Programmers's Reference*; Wrox Press, Birmingham; 2000; ISBN 1-861003-12-9
- **Ray, Erik T.:** *Einführung in XML*; O'Reilly, Köln; 2001; ISBN 3-89721-286-2
- **Tidwell, Doug:** *XSLT. XML-Dokumente transformieren*; O'Reilly, Köln; 2002; ISBN 3-89721-292-7

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (d  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

- 
- *Bücher zu XML und angrenzenden Gebieten;*  
<http://www.uni-giessen.de/partosch/xml/buecher.html>  
(nicht mehr gepflegt, da das Buchangebot mittlerweile zu umfangreich und unübersichtlich geworden ist)

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

# 10 URLs

- Apache XML Project: *FOP*  
<http://xml.apache.org/fop/>
- Diskussionsliste [xml-de@listserv.dfn.de](mailto:xml-de@listserv.dfn.de)  
<mailto:xml-de@listserv.gmd.de>
- Diskussionsliste [xml-l@LISTSERV.HEANET.IE](mailto:xml-l@LISTSERV.HEANET.IE)  
<mailto:xml-l@LISTSERV.HEANET.IE>
- DocBook TC : *DocBook*  
<http://www.oasis-open.org/docbook/>
- Lars Marius Garshol: *Free XML tools and software*  
<http://www.garshol.priv.no/download/xmltools/>
- Hans Hagen: *MAPS bibliografie*  
<http://www.pragma-ade.com/itdemos/internal/maps/maps.pdf>
- Hans Hagen: *MATHML*  
<http://www.pragma-ade.nl/general/manuals/xmathml-p.pdf>

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (1  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

- Hans Hagen: *MathML in ConT<sub>E</sub>Xt*  
<http://www.pragma-ade.nl/general/manuals/mmlexamp.pdf>
- Hans Hagen: *MathML*  
<http://www.pragma-ade.nl/general/manuals/mmlprime.pdf>
- P. Hoffman, F. Yergeau: **UTF-16**  
<http://www.ietf.org/rfc/rfc2781.txt>
- Michael H. Kay: *DTDGenerator – A tool to generate XML DTDs*  
<http://saxon.sourceforge.net/dtdgen.html>
- Microsoft: *XML Tutorial*  
URL zu lang!
- Microsoft: *XSLT Developer's Guide*  
URL zu lang!
- Günter Partosch: *Einführung in die eXtensible Markup Language (XML)*  
<http://www.uni-giessen.de/partosch/xml/kurs/>
- Günter Partosch: *Einführung in XML – Konzepte, Arbeitsabläufe*  
<http://www.uni-giessen.de/partosch/xml/kurs2/>

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

- Günter Partosch: *Bücher zu XML und angrenzenden Gebieten*  
<http://www.uni-giessen.de/partosch/xml/buecher.html>
- Dave Raggett: *Clean up your Web pages with HTML TIDY*  
<http://www.w3.org/People/Raggett/tidy/>
- Text Encoding Initiative: *Text Encoding Initiative*  
<http://www.tei-c.org/>
- Usenet-News-Gruppe `comp.text.xml`  
`news:comp.text.xml`
- Usenet-News-Gruppe `de.comp.text.xml`  
`news:de.comp.text.xml`
- Usenet-News-Gruppe `microsoft.de.german.public.xml`  
`news:microsoft.de.german.public.xml`
- Usenet-News-Gruppe `microsoft.public.xml`  
`news:microsoft.public.xml`
- Usenet-News-Gruppe `microsoft.public.xsl`  
`news:microsoft.public.xsl`

Ausgangsposition um  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch



- W3C: *W3C's Math Home Page*  
<http://www.w3.org/Math/>
- W3C: *Document Object Model (DOM) Level 2 Core Specification*  
<http://www.w3.org/TR/DOM-Level-2-Core>
- W3C: *Document Object Model (DOM) Level 2 Events Specification*  
<http://www.w3.org/TR/DOM-Level-2-Events>
- W3C: *Document Object Model (DOM) Level 2 Style Specification*  
<http://www.w3.org/TR/DOM-Level-2-Style>
- W3C: *Document Object Model (DOM) Level 2 Traversal and Range Specification*  
<http://www.w3.org/TR/DOM-Level-2-Traversal-Range>
- W3C: *Document Object Model (DOM) Level 2 Views Specification*  
<http://www.w3.org/TR/DOM-Level-2-Views>
- W3C: *Cascading Style Sheets, level 2 (CSS2) Specification*  
<http://www.w3.org/TR/CSS2>
- W3C: *Document Object Model (DOM) Level 1*  
<http://www.w3.org/TR/DOM-Level-1>

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

**Zurück**

**Inhalt**

**Erste**

**Vorherige**

**Nächste**

**Letzte**

- W3C: *Mathematical Markup Language (MathML) 1.01 Specification*  
<http://www.w3.org/TR/REC-MathML/>
- W3C: *Namespaces in XML*  
<http://www.w3.org/TR/xml-names>
- W3C: *XHTML 1.0: The Extensible HyperText Markup Language – A Re-formulation of HTML 4 in XML 1.0*  
<http://www.w3.org/TR/xhtml1/>
- W3C: *XML Linking Language (XLink) Version 1.0*  
<http://www.w3.org/TR/xlink/>
- W3C: *XML Base*  
<http://www.w3.org/TR/xmlbase/>
- W3C: *XML Information Set (Second Edition)*  
<http://www.w3.org/TR/xml-infoset>
- W3C: *XML Schema Part 0: Primer*  
<http://www.w3.org/TR/xmlschema-0/>
- W3C: *XML Schema Part 1: Structures*  
<http://www.w3.org/TR/xmlschema-1/>

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen (URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

- W3C: *XML Schema Part 2: Datatypes*  
<http://www.w3.org/TR/xmlschema-2/>
- W3C: *Associating Style Sheets with XML documents*  
<http://www.w3.org/TR/xml-styleSheet>
- W3C: *Extensible Stylesheet Language (XSL) Version 1.0*  
<http://www.w3.org/TR/xsl/>
- W3C: *Cascading Style Sheets (CSS1) Level 1 Specification*  
<http://www.w3.org/TR/CSS1>
- W3C: *Extensible Markup Language (XML) 1.0 (Third Edition)*  
<http://www.w3.org/TR/xml>
- W3C: *Extensible Markup Language (XML) 1.1*  
<http://www.w3.org/TR/xml11/>
- W3C: *Canonical XML Version 1.0*  
<http://www.w3.org/TR/xml-c14n>
- W3C: *XML Path Language (XPath) Version 1.0*  
<http://www.w3.org/TR/xpath>

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

- W3C: *XSL Transformations (XSLT) Version 1.0*  
<http://www.w3.org/TR/xslt>
- F. Yergeau: **UTF-8**  
<http://www.ietf.org/rfc/rfc2279.txt>

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

Suchen

Abbruch

Zurück

Inhalt

Erste

Vorherige

Nächste

Letzte

# 11 Nachtrag: XSL Formatting Objects

- Mit Hilfe der *XSL Formatting Objects* werden relativ detailliert Layout und Formatierung eines zu setzenden Dokuments festgelegt.
- Dateien mit Formatting Objects sind XML-Dateien. Sie werden normalerweise durch XSLT-Prozessoren erzeugt.
- FO-Prozessoren lesen diese Dateien und setzen das Dokument.
- Beispiele für FO-Prozessoren:
  - *FOP*
  - PassiveT<sub>E</sub>X

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Üb  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

## Beispiel 1 (Äußeres Gerüst)

Das folgende XSLT-Fragment könnte Formatting Objects erzeugen, die das globale Layout eines Dokuments festlegen:

```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/XSL/Transform/1.0"
                xmlns:fo="http://www.w3.org/XSL/Format/1.0"
                default-space=""><
<xsl:variable name="PageMarginTop">75pt</xsl:variable>
<xsl:variable name="PageMarginBottom">125pt</xsl:variable>
<xsl:variable name="PageMarginLeft">80pt</xsl:variable>
<xsl:variable name="PageMarginRight">150pt</xsl:variable>
<xsl:variable name="BodySize">12pt</xsl:variable>

<xsl:template match='/'>
  <fo:root xmlns:fo="http://www.w3.org/XSL/Format/1.0">
    <fo:layout-master-set>
      <fo:simple-page-master
        page-master-name="allpages"
        margin-top="{ $PageMarginTop }"
        margin-bottom="{ $PageMarginBottom }"
        margin-left="{ $PageMarginLeft }"
        margin-right="{ $PageMarginRight }">
```

Ausgangsposition un  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Übe  
Arbeitsabläufe – Beis  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**

```

        <fo:region-body margin-bottom="100pt"/>
        <fo:region-after extent="25pt"/>
    </fo:simple-page-master>
</fo:layout-master-set>
<fo:page-sequence>
    <fo:sequence-specification>
        <fo:sequence-specifier-repeating
            page-master-first="allpages"
            page-master-repeating="allpages"/>
    </fo:sequence-specification>
    <fo:flow font-family="serif">
        <xsl:apply-templates/>
    </fo:flow>
</fo:page-sequence>
</fo:root>
</xsl:template>

...
</xsl:stylesheet>

```

Ausgangsposition und  
 Markup  
 SGML  
 XML  
 XML – Begleitende  
 Standardisierte XML  
 Arbeitsabläufe – Über  
 Arbeitsabläufe – Beis  
 Informationsquellen  
 URLs  
 Nachtrag: XSL Form

Suchen

Abbruch

## Beispiel 2 (Aufzählung)

XML-Fragment aus einem XML-Dokument:

```
<ol>
<li>Punkt 1</li>
<li>Punkt 2</li>
</ol>
```

Die zugehörigen Formatting Objects könnten (verkürzt) durch das folgende XSLT-Fragment generiert werden:

```
<xsl:template match="ol">
  <fo:list-block space-before.optimum="4pt">
    <xsl:apply-templates />
  </fo:list-block>
</xsl:template>
```

und

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

**Suchen**

**Abbruch**



```
<xsl:template match="li">
  <fo:list-item>
    <fo:list-item-label>...</fo:list-item-label>
    <fo:list-item-body>
      <fo:block font-weight="normal">
        <xsl:apply-templates />
      </fo:block>
    </fo:list-item-body>
  </fo:list-item>
</xsl:template>
```

Ausgangsposition und  
Markup  
SGML  
XML  
XML – Begleitende  
Standardisierte XML  
Arbeitsabläufe – Über  
Arbeitsabläufe – Bei  
Informationsquellen  
URLs  
Nachtrag: XSL Form

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Inhalt](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

tutc7.dir

```
\def\texttt#1{{\tt #1}}
\def\adresse#1{\subsubject{Adresse~Nr.~#1}}
\def\startanrede#1\stopanrede{#1}
\def\starttitel#1\stoptitel{#1}
\def\startvorname#1\stopvorname{#1}
\def\startnachname#1\stopnachname{#1\par}
\def\startsname#1\stopsname{#1}
\def\starthnr#1\stophnr{#1\par}
\def\startpostfach#1\stoppostfach{Postfach:~#1\par}
\def\startplz#1\stopplz{#1}
\def\startortsname#1\stoportsname{#1\par\smallskip}
\def\starttelefon#1\stoptelefon{Telefon:~#1\par}
\def\startfax#1\stopfax{Fax:~#1\par}
\def\startemail#1\stopemail{E-Mail:~\texttt{#1}}

\defineXMLenvironment [adressen]
  {\useencoding[win]\starttext\title{Adressen}}
  {\stoptext}
\defineXMLpickup [adresse]
  {\adresse{\XMLpar{adresse}{nr}{--}}}
  {}
```

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

<code>\defineXMLpickup</code>	<code>[anrede]</code>	<code>\startanrede</code>	<code>\stopanrede</code>
<code>\defineXMLpickup</code>	<code>[titel]</code>	<code>\starttitel</code>	<code>\stoptitel</code>
<code>\defineXMLpickup</code>	<code>[vorname]</code>	<code>\startvorname</code>	<code>\stopvorname</code>
<code>\defineXMLpickup</code>	<code>[nachname]</code>	<code>\startnachname</code>	<code>\stopnachname</code>
<code>\defineXMLpickup</code>	<code>[sname]</code>	<code>\startsname</code>	<code>\stopsname</code>
<code>\defineXMLpickup</code>	<code>[hnr]</code>	<code>\starthnr</code>	<code>\stophnr</code>
<code>\defineXMLpickup</code>	<code>[postfach]</code>	<code>\startpostfach</code>	<code>\stoppostfach</code>
<code>\defineXMLpickup</code>	<code>[plz]</code>	<code>\startplz</code>	<code>\stopplz</code>
<code>\defineXMLpickup</code>	<code>[ortsname]</code>	<code>\startortsname</code>	<code>\stoportsname</code>
<code>\defineXMLpickup</code>	<code>[telefon]</code>	<code>\starttelefon</code>	<code>\stoptelefon</code>
<code>\defineXMLpickup</code>	<code>[fax]</code>	<code>\startfax</code>	<code>\stopfax</code>
<code>\defineXMLpickup</code>	<code>[email]</code>	<code>\startemail</code>	<code>\stopemail</code>

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

adr6.xml

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <!-- adr6.xml -->
3  <?xml-stylesheet href="adr2.xsl" type="text/xsl" ?>
4  <!DOCTYPE adressen SYSTEM "adressen.dtd" >
5  <adressen>
6
7      <adresse nr="1">
8          <anrede>Frau</anrede>
9          <titel>Dr.</titel>
10         <name>
11             <vorname>Anna</vorname>
12             <nachname>Mayer</nachname>
13         </name>
14         <strasse>
15             <sname>Blumenweg</sname>
16             <hnr>2</hnr>
17         </strasse>
18         <ort>
19             <plz>35392</plz>
20             <ortsname>Gießen</ortsname>
21         </ort>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
21 <telefon>0641-123456</telefon>
22 <fax>0641-123459</fax>
23 <email>Anna.Mayer@t-online.de</email>
24 </adresse>

25 <adresse nr="2">
26   <anrede>Herr</anrede>
27   <name>
28     <vorname>Emil</vorname>
29     <nachname>Müller</nachname>
30   </name>
31   <strasse>
32     <sname>Tulpenweg</sname>
33     <hnr>22a</hnr>
34   </strasse>
35   <ort>
36     <plz>35398</plz>
37     <ortsname>Gießen</ortsname>
38   </ort>
39   <telefon>0641-234567</telefon>
40   <email>Emil.Mueller@profanistik.uni-giessen.de</email>
41 </adresse>
```

Suchen

Abbruch

```
42 <adresse nr="3">
43   <anrede>Frau</anrede>
44   <titel>Prof. Dr.</titel>
45   <name>
46     <vorname>Erna</vorname>
47     <nachname>Schneider</nachname>
48   </name>
49   <strasse>
50     <sname>Bachstraße</sname>
51     <hnr>10</hnr>
52   </strasse>
53   <postfach>12345</postfach>
54   <ort>
55     <plz>12340</plz>
56     <ortsname>Nixdorf</ortsname>
57   </ort>
58   <telefon>09994-1234</telefon>
59   <fax>09994-12349</fax>
60   <email>Erna.Schneider@gmx.de</email>
61 </adresse>
62 </adressen>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

adressen.xmt

```
\XMLElement{adressen}{  
  {\documentclass[10pt,a4paper]{article}  
  \usepackage{adr-tex}  
  \title{Adressen}  
  \author{Emil Mayer}  
  \date{\today}  
  \begin{document}  
  \maketitle  
  \section*{Adressen}  
  }  
  {\end{document}}}  
XMLElement{adresse}{\XMLAttribute{nr}{\nummer}{--}}  
  {\begin{adresse}{\nummer}}  
  {\end{adresse}}  
  
XMLElement{anrede}    {} {\xmlgrab} {\anrede{#1}}  
XMLElement{titel}     {} {\xmlgrab} {\titel{#1}}  
XMLElement{vorname}   {} {\xmlgrab} {\vorname{#1}}  
XMLElement{nachname}  {} {\xmlgrab} {\nachname{#1}}  
XMLElement{sname}     {} {\xmlgrab} {\sname{#1}}  
XMLElement{hnr}       {} {\xmlgrab} {\hnr{#1}}
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
\XMLelement{postfach} {} {\xmlgrab} {\postfach{#1}}  
\XMLelement{plz}      {} {\xmlgrab} {\plz{#1}}  
\XMLelement{ortsname} {} {\xmlgrab} {\ortsname{#1}}  
\XMLelement{telefon}  {} {\xmlgrab} {\telefon{#1}}  
\XMLelement{fax}      {} {\xmlgrab} {\fax{#1}}  
\XMLelement{email}    {} {\xmlgrab} {\email{#1}}
```

**Suchen**

**Abbruch**

**Zurück**

**Erste**

**Vorherige**

**Nächste**

**Letzte**



adr6.xml

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <!-- adr6.xml -->
3  <?xml-stylesheet href="adr2.xsl" type="text/xsl" ?>
4  <!DOCTYPE adressen SYSTEM "adressen.dtd" >
5  <adressen>

6  <adresse nr="1">
7    <anrede>Frau</anrede>
8    <titel>Dr.</titel>
9    <name>
10     <vorname>Anna</vorname>
11     <nachname>Mayer</nachname>
12   </name>
13   <strasse>
14     <sname>Blumenweg</sname>
15     <hnr>2</hnr>
16   </strasse>
17   <ort>
18     <plz>35392</plz>
19     <ortsname>Gießen</ortsname>
20   </ort>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
21      <telefon>0641-123456</telefon>
22      <fax>0641-123459</fax>
23      <email>Anna.Mayer@t-online.de</email>
24      </adresse>

25  <adresse nr="2">
26      <anrede>Herr</anrede>
27      <name>
28          <vorname>Emil</vorname>
29          <nachname>Müller</nachname>
30      </name>
31      <strasse>
32          <sname>Tulpenweg</sname>
33          <hnr>22a</hnr>
34      </strasse>
35      <ort>
36          <plz>35398</plz>
37          <ortsname>Gießen</ortsname>
38      </ort>
39      <telefon>0641-234567</telefon>
40      <email>Emil.Mueller@profanistik.uni-giessen.de</email>
41      </adresse>
```

Suchen

Abbruch

```
42 <adresse nr="3">
43   <anrede>Frau</anrede>
44   <titel>Prof. Dr.</titel>
45   <name>
46     <vorname>Erna</vorname>
47     <nachname>Schneider</nachname>
48   </name>
49   <strasse>
50     <sname>Bachstraße</sname>
51     <hnr>10</hnr>
52   </strasse>
53   <postfach>12345</postfach>
54   <ort>
55     <plz>12340</plz>
56     <ortsname>Nixdorf</ortsname>
57   </ort>
58   <telefon>09994-1234</telefon>
59   <fax>09994-12349</fax>
60   <email>Erna.Schneider@gmx.de</email>
61 </adresse>
62 </adressen>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

## kapitel2.xml

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <?xml-stylesheet href="kapitel2-tex1.xsl" type="text/xsl" ?>
3  <!DOCTYPE mehrerekapitel SYSTEM "kapitel2.dtd" >

4  <!-- kapitel2.xml -->
5  <mehrerekapitel>
6  <einkapitel>
7      <titel>Titel des ersten Kapitels</titel>
8      <rumpf>
9          <absatz>Ein bisschen Text ...</absatz>
10         <absatz>Noch ein bisschen Text ... </absatz>
11     </rumpf>
12 </einkapitel>

13 <einkapitel>
14     <titel>Titel des zweiten Kapitels</titel>
15     <rumpf>
16         <absatz>Ein bisschen anderer Text ...</absatz>
17         <absatz>Noch ein bisschen anderer Text ... </absatz>
18     </rumpf>
19 </einkapitel>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte



kap2t.awk

```
# kapitel1-html1.awk
```

```
BEGIN      {print "\\documentclass[a4paper,10pt]{article}"
            print " "
            print "\\usepackage[latin1]{inputenc}"
            print "\\usepackage{ngerman}"
            print "\\title{Unser Kapitel}"
            print "\\author{Emil Mayer}"
            print "\\date{\\today}"
            print " "
            print "\\begin{document}"
            print "\\maketitle"
            intitel=0;titel=""
            imabsatz=0;absatz=""}
```

```
/<mehrerekapitel>/      {next}
/<[\\/]mehrerekapitel>/  {next}
```

```
/<einkapitel>/          {next}
/<[\\/]einkapitel>/      {next}
```

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

<code>/&lt;titel&gt;/</code>	<code>{intitel=1;next}</code>
<code>/&lt;[\/]titel&gt;/</code>	<code>{printf("\n")</code>
	<code>print "\\section{" titel "}"</code>
	<code>printf("\n")</code>
	<code>intitel=0;titel="";next}</code>
<code>intitel==1</code>	<code>{titel=titel \$0;next}</code>
<code>/&lt;absatz&gt;/</code>	<code>{imabsatz=1;next}</code>
<code>/&lt;[\/]absatz&gt;/</code>	<code>{print absatz</code>
	<code>print " "</code>
	<code>imabsatz=0;absatz="";next}</code>
<code>imabsatz==1</code>	<code>{absatz=absatz \$0;next}</code>
<code>/&lt;rumpf&gt;/</code>	<code>{next}</code>
<code>/&lt;[\/]rumpf&gt;/</code>	<code>{next}</code>
<code>END</code>	<code>{print "\\end{document}"}</code>

[Suchen](#)

[Abbruch](#)

## kap2h.awk

```
# kapitel1-html1.awk
BEGIN          {print "<html>"
                print "<head><title>Unsere Kapitel</title></head>"
                print "<body>"
                print "<h1>Unsere Kapitel</h1>"
                print " "
                intitel=0;titel=""
                imabsatz=0;absatz=""}

/<mehrerekapitel>/      {next}
/<[\\/]mehrerekapitel>/ {next}

/<einkapitel>/          {next}
/<[\\/]einkapitel>/      {next}

/<titel>/               {intitel=1;next}
/<[\\/]titel>/           {printf("\n")
                        print "<h2>" titel "</h2>"
                        printf("\n")
                        intitel=0;titel="";next}

intitel==1             {titel=titel $0;next}
```

[Suchen](#)

[Abbruch](#)



<code>/&lt;absatz&gt;/</code>	<code>{imabsatz=1;next}</code>
<code>/&lt;[\\/]absatz&gt;/</code>	<code>{print "&lt;p&gt;" absatz "&lt;/p&gt;"</code>
	<code>imabsatz=0;absatz="";next}</code>
<code>imabsatz==1</code>	<code>{absatz=absatz \$0;next}</code>
<code>/&lt;rumpf&gt;/</code>	<code>{next}</code>
<code>/&lt;[\\/]rumpf&gt;/</code>	<code>{next}</code>
<code>END</code>	<code>{print "&lt;/body&gt;"</code>
	<code>print "&lt;/html&gt;"}</code>

[Suchen](#)

[Abbruch](#)

[Zurück](#)



[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

kapitel2.tex

```
\documentclass[a4paper,10pt]{article}
```

```
\usepackage[latin1]{inputenc}
```

```
\usepackage{ngerman}
```

```
\title{Unser Kapitel}
```

```
\author{Emil Mayer}
```

```
\date{\today}
```

```
\begin{document}
```

```
\maketitle
```

```
\section{Titel des ersten Kapitels}
```

Ein bisschen Text ...

Noch ein bisschen Text ...

```
\section{Titel des zweiten Kapitels}
```

Ein bisschen anderer Text ...

Noch ein bisschen anderer Text ...

\end{document}

Zurück



Erste

Vorherige

Nächste

Letzte

Suchen

Abbruch

kapitel2.html

```
<html>
<head><title>Unsere Kapitel</title></head>
<body>
<h1>Unsere Kapitel</h1>

<h2>Titel des ersten Kapitels</h2>

<p>Ein bisschen Text ...</p>
<p>Noch ein bisschen Text ... </p>

<h2>Titel des zweiten Kapitels</h2>

<p>Ein bisschen anderer Text ...</p>
<p>Noch ein bisschen anderer Text ... </p>
</body>
</html>
```

[Suchen](#)

[Abbruch](#)

[Zurück](#)



[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

zerteil.awk

```
# pr0.awk
```

```
BEGIN      {zk="<[\\/?][a-z]*[a-zA-Z\\\"=]*>"}
```

```
$0~zk      {sub (/^[ \t]*/, "", $0)
```

```
            while (match ($0,zk) )
```

```
                {if (RSTART>1) print substr($0, 1, RSTART-1)
```

```
                print substr($0, RSTART, RLENGTH)
```

```
                $0=substr($0, RSTART + RLENGTH)}
```

```
            next}
```

```
/^[ \t]*$/ {next}
```

```
{print}
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

## kapitel2-tex1.xsl

```
1  <?xml version='1.0' encoding="ISO-8859-1" ?>

2  <!-- kapitel2-tex1.xsl
3      XSL-Datei (passend zu kapitel2.xml
4      kann mit SAXON und anderen Tools verarbeitet werden
5      28.1.2002, GP (HRZ Gießen)                                -->

6  <xsl:stylesheet version="1.0"
7      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
8      <xsl:output method="text" encoding="ISO-8859-1" />
9      <xsl:template match="/">
10         <xsl:text>\documentclass[a4paper,10pt]{article}

11         \usepackage[latin1]{inputenc}
12         \usepackage{ngerman}
13         \title{Unser Kapitel}
14         \author{Emil Mayer}
15         \date{\today}

16         \begin{document}
17         \maketitle</xsl:text>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
18     <xsl:apply-templates />
19     <xsl:text>\end{document}</xsl:text>
20 </xsl:template>

21 <xsl:template match="kapitel">
22     <xsl:apply-templates />
23 </xsl:template>

24 <xsl:template match="titel">
25     <xsl:text>\section{</xsl:text>
26     <xsl:apply-templates />
27     <xsl:text>}</xsl:text>
28 </xsl:template>

29 <xsl:template match="rumpf">
30     <xsl:apply-templates />
31 </xsl:template>

32 <xsl:template match="absatz">
33     <xsl:apply-templates />
34     <xsl:text>\par</xsl:text>
35 </xsl:template>
```

Suchen

Abbruch

```
36     <xsl:template match="text()">
37         <xsl:value-of select="." />
38     </xsl:template>
39 </xsl:stylesheet>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte



kapitel2a.tex

```
\documentclass[a4paper,10pt]{article}
```

```
\usepackage[latin1]{inputenc}
```

```
\usepackage{ngerman}
```

```
\title{Unser Kapitel}
```

```
\author{Emil Mayer}
```

```
\date{\today}
```

```
\begin{document}
```

```
\maketitle
```

```
\section{Titel des ersten Kapitels}
```

```
Ein bisschen Text ...\par
```

```
Noch ein bisschen Text ... \par
```

```
\section{Titel des zweiten Kapitels}
```

```
Ein bisschen anderer Text ...\par
```

```
Noch ein bisschen anderer Text ... \par
```

[Suchen](#)

[Abbruch](#)

[Zurück](#)



[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

\end{document}

Zurück



Erste

Vorherige

Nächste

Letzte

Suchen

Abbruch

## kapitel2-html1.xsl

```
1  <?xml version='1.0' encoding="ISO-8859-1" ?>

2  <!-- kapitel2-tex1.xsl
3      XSL-Datei (passend zu kapitel2.xml
4      kann mit SAXON und anderen Tools verarbeitet werden
5      28.1.2002, GP (HRZ Gießen)                                -->

6  <xsl:stylesheet version="1.0"
7  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
8      <xsl:output method="html" encoding="ISO-8859-1" />
9      <xsl:template match="/">
10         <html>

11             <head><title>Unsere Kapitel</title></head>

12             <body>
13                 <h1>Unsere Kapitel</h1>
14                 <xsl:apply-templates />
15             </body>
16         </html>
17     </xsl:template>
```

Suchen

Abbruch

```
18 <xsl:template match="kapitel">
19     <xsl:apply-templates />
20 </xsl:template>

21 <xsl:template match="titel">
22     <h2><xsl:apply-templates /></h2>
23 </xsl:template>

24 <xsl:template match="rumpf">
25     <xsl:apply-templates />
26 </xsl:template>

27 <xsl:template match="absatz">
28     <p><xsl:apply-templates /></p>
29 </xsl:template>

30 <xsl:template match="text(">
31     <xsl:value-of select="." />
32 </xsl:template>
33 </xsl:stylesheet>
```

Suchen

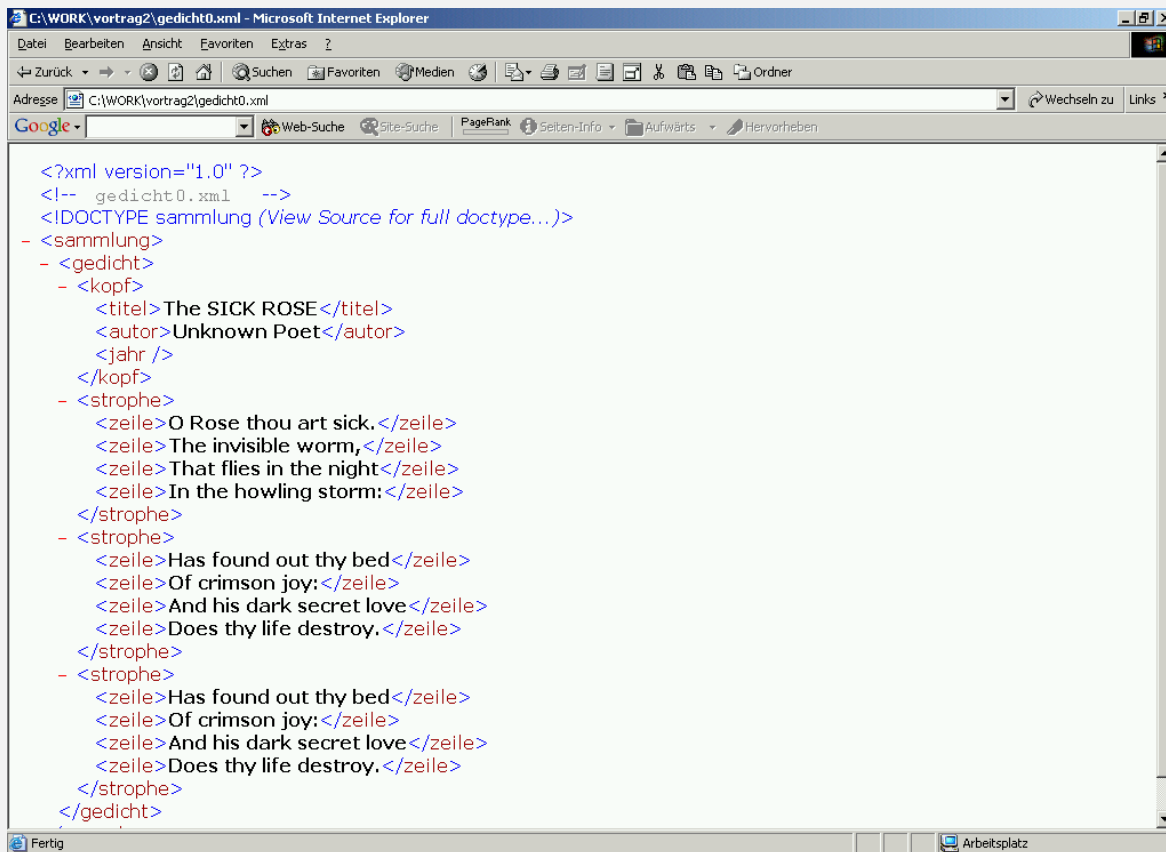
Abbruch

```
1  <html>
2    <head>
3      <meta http-equiv="Content-Type" content="text/html; charset=ISO-
4  8859-1">
5
6    <title>Unsere Kapitel</title>
7  </head>
8  <body>
9    <h1>Unsere Kapitel</h1>
10
11   <h2>Titel des ersten Kapitels</h2>
12
13   <p>Ein bisschen Text ...</p>
14
15   <p>Noch ein bisschen Text ... </p>
16
17   <h2>Titel des zweiten Kapitels</h2>
18
19   <p>Ein bisschen anderer Text ...</p>
20
21   <p>Noch ein bisschen anderer Text ... </p>
```

[Suchen](#)

[Abbruch](#)

```
15     </body>
16 </html>
```



**Abbildung 1** Darstellung der XML-Datei gedicht0.xml im Internet Explorer mittels des Default-Style-Sheets

Suchen

Abbruch

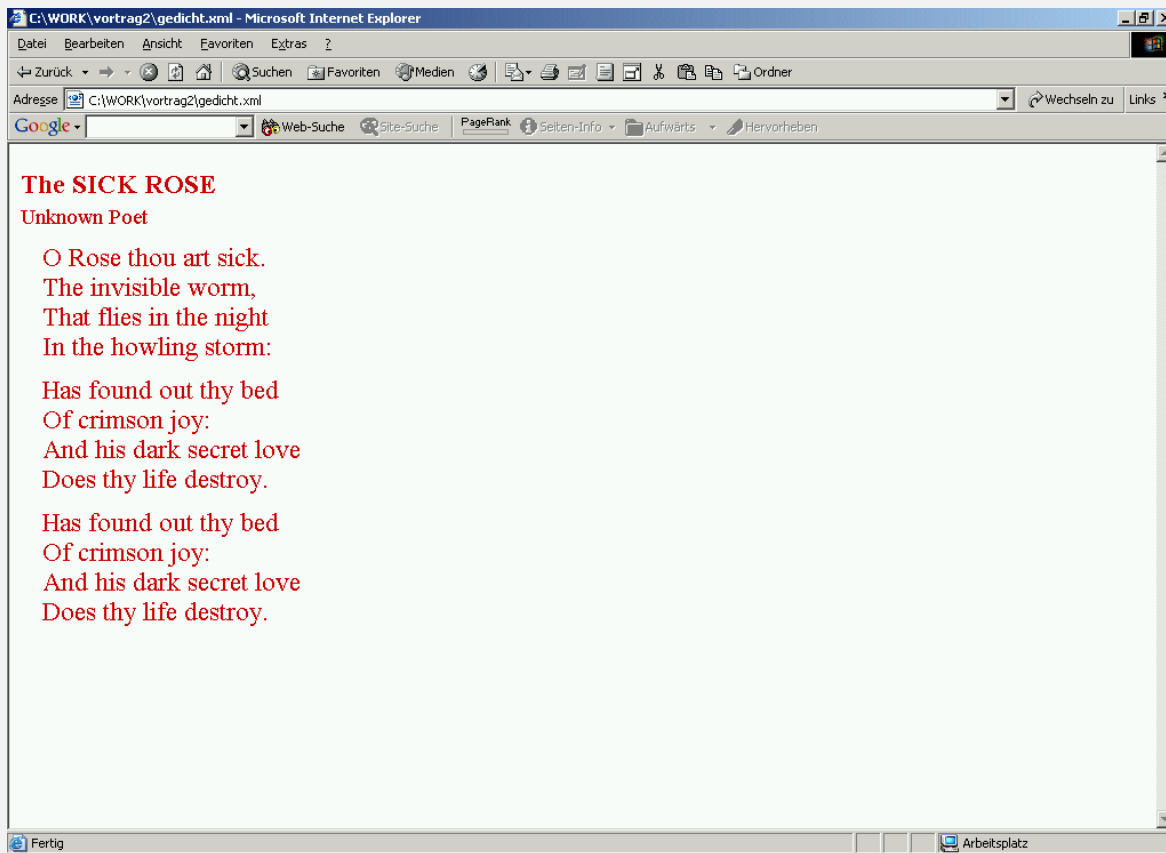
Zurück

Erste

Vorherige

Nächste

Letzte



**Abbildung 1** Darstellung der XML-Datei gedicht.xml im Internet Explorer mittels der CSS-Datei gedicht.css

Suchen

Abbruch

Zurück

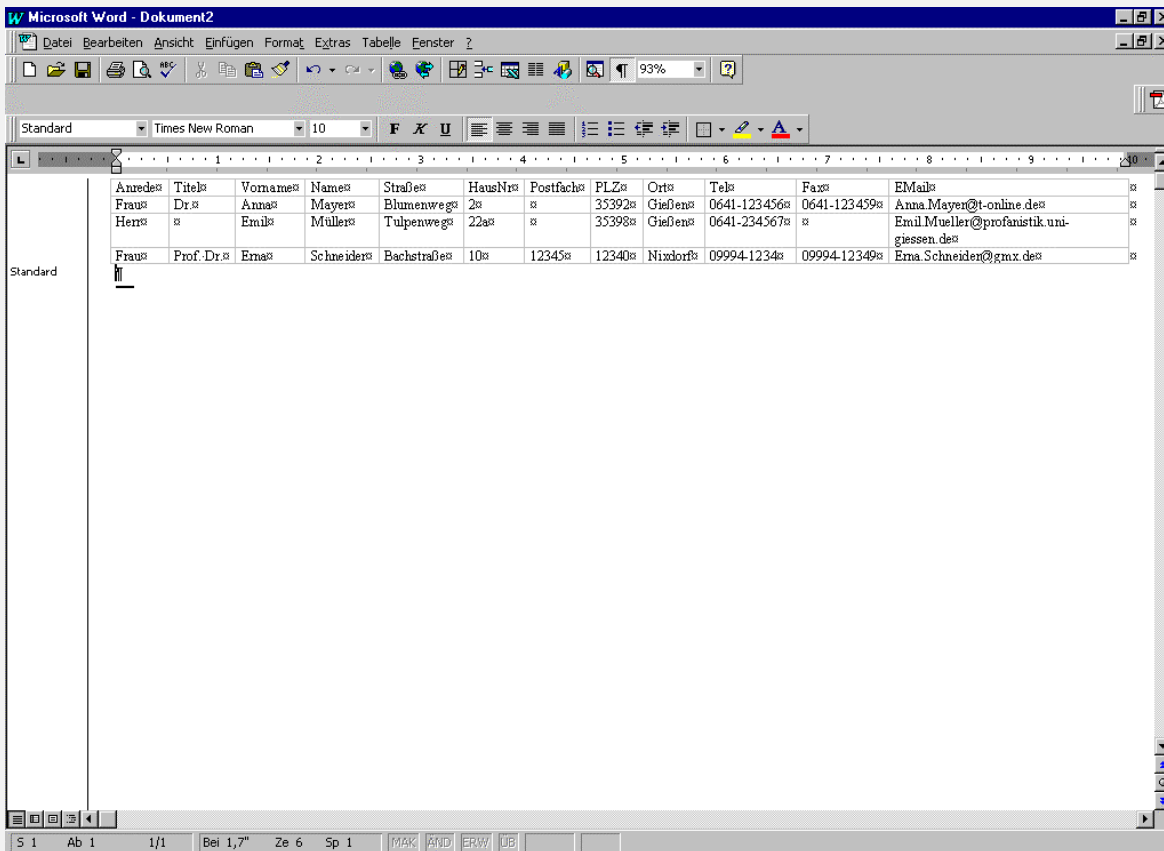
Erste

Vorherige

Nächste

Letzte





**Abbildung 1** Ausgangstabelle in WinWord

Suchen

Abbruch

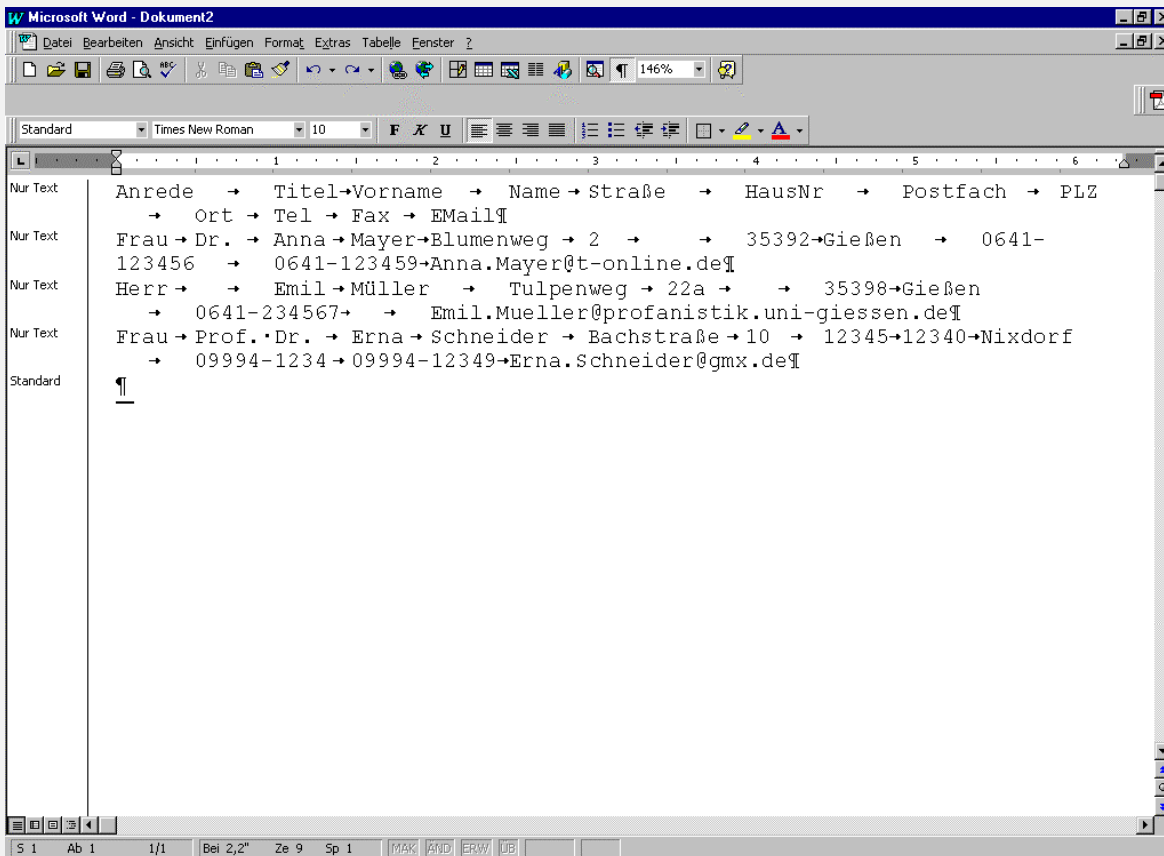
Zurück

Erste

Vorherige

Nächste

Letzte



**Abbildung 1** ASCII-Rohfassung adr.txt der Tabelle in WinWord

Suchen

Abbruch

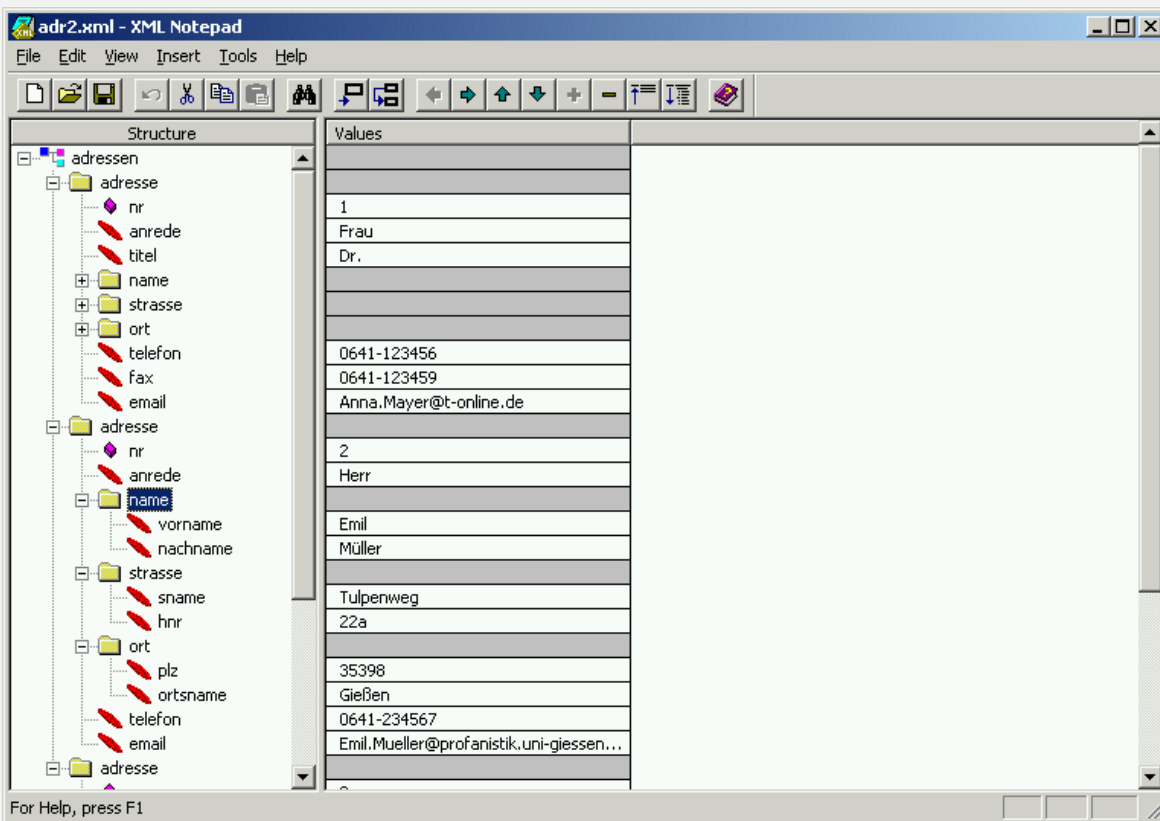
Zurück

Erste

Vorherige

Nächste

Letzte



**Abbildung 1** Darstellung der XML-Datei adr2.xml im XML Notepad

Suchen

Abbruch

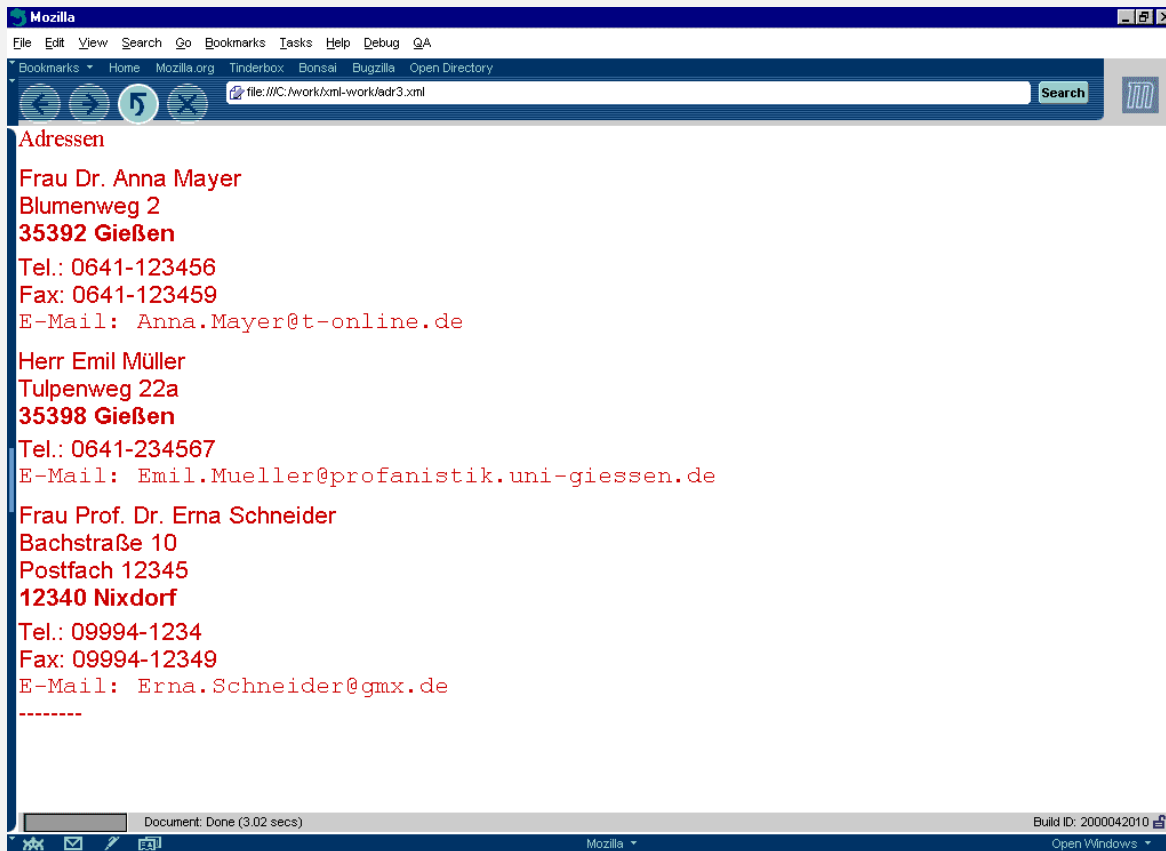
Zurück

Erste

Vorherige

Nächste

Letzte



**Abbildung 1** Darstellung der XML-Datei adr3.xml im Mozilla mittels der CSS-Datei adr.css

Suchen

Abbruch

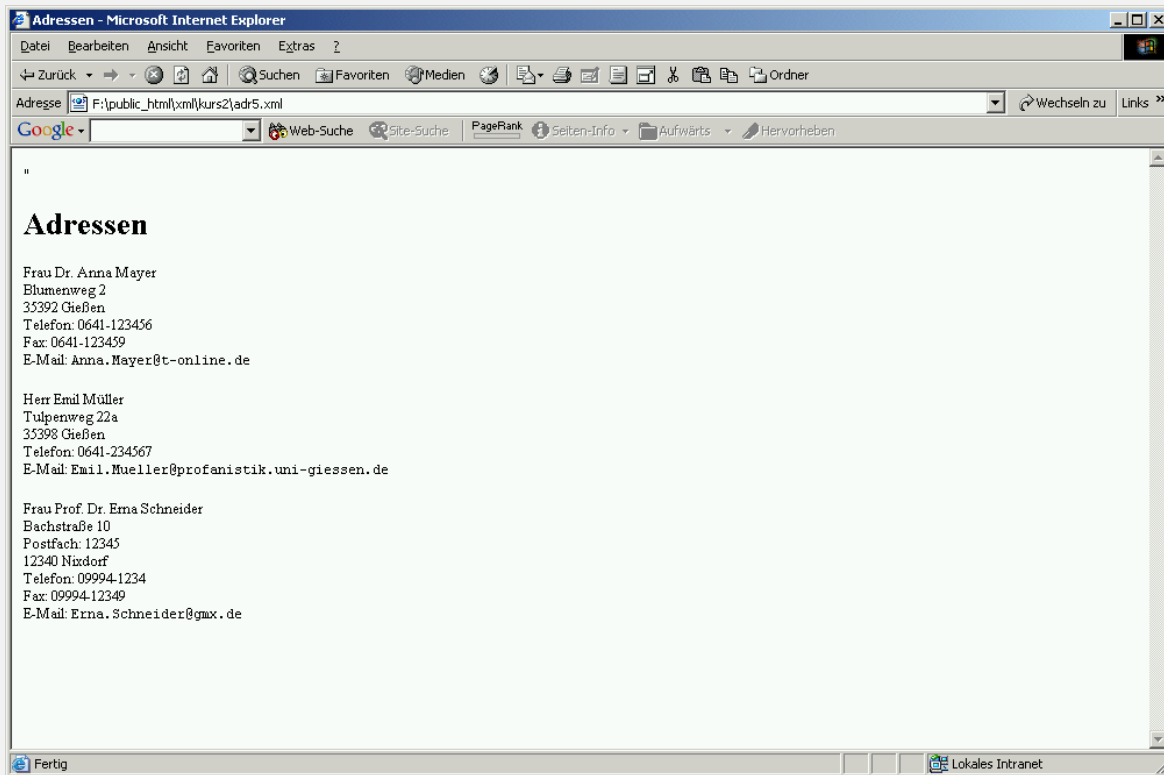
Zurück

Erste

Vorherige

Nächste

Letzte



**Abbildung 1** Darstellung der XML-Datei adr5.xml im Internet Explorer mittels der XSL-Datei adr.xsl

Suchen

Abbruch

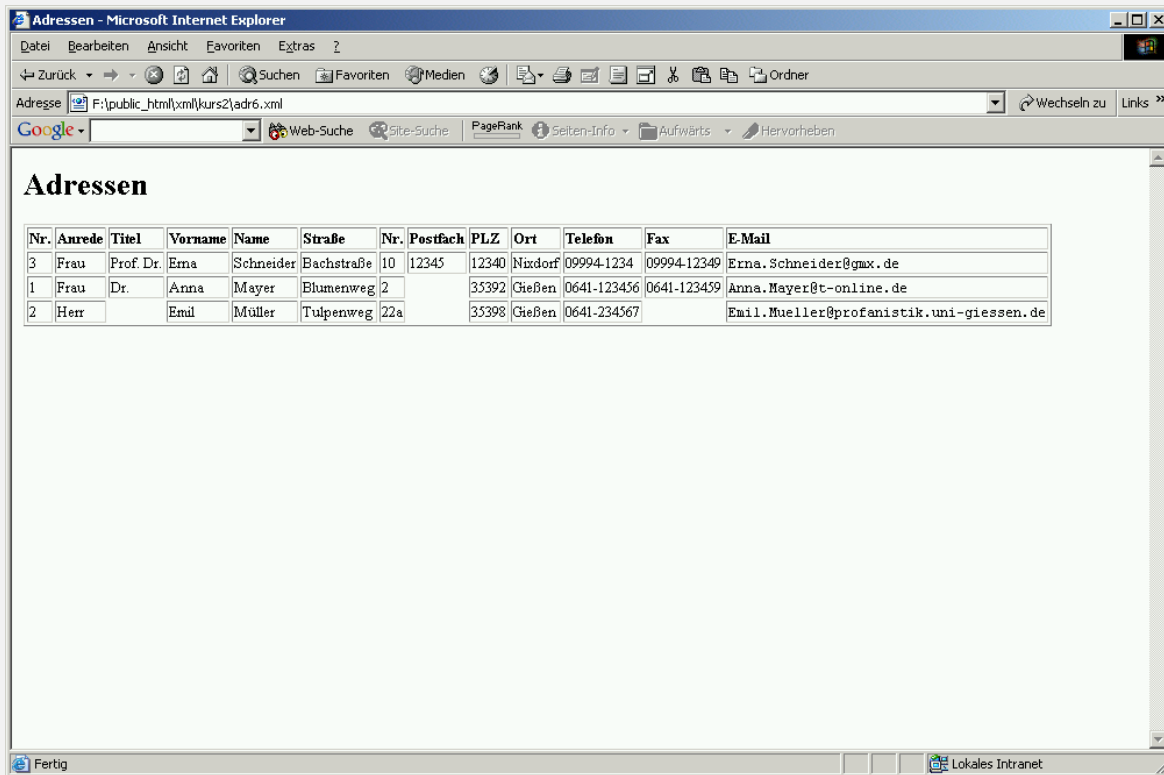
Zurück

Erste

Vorherige

Nächste

Letzte



**Abbildung 1** Darstellung der XML-Datei adr6.xml im Internet Explorer mittels der XSL-Datei adr2.xsl

Suchen

Abbruch

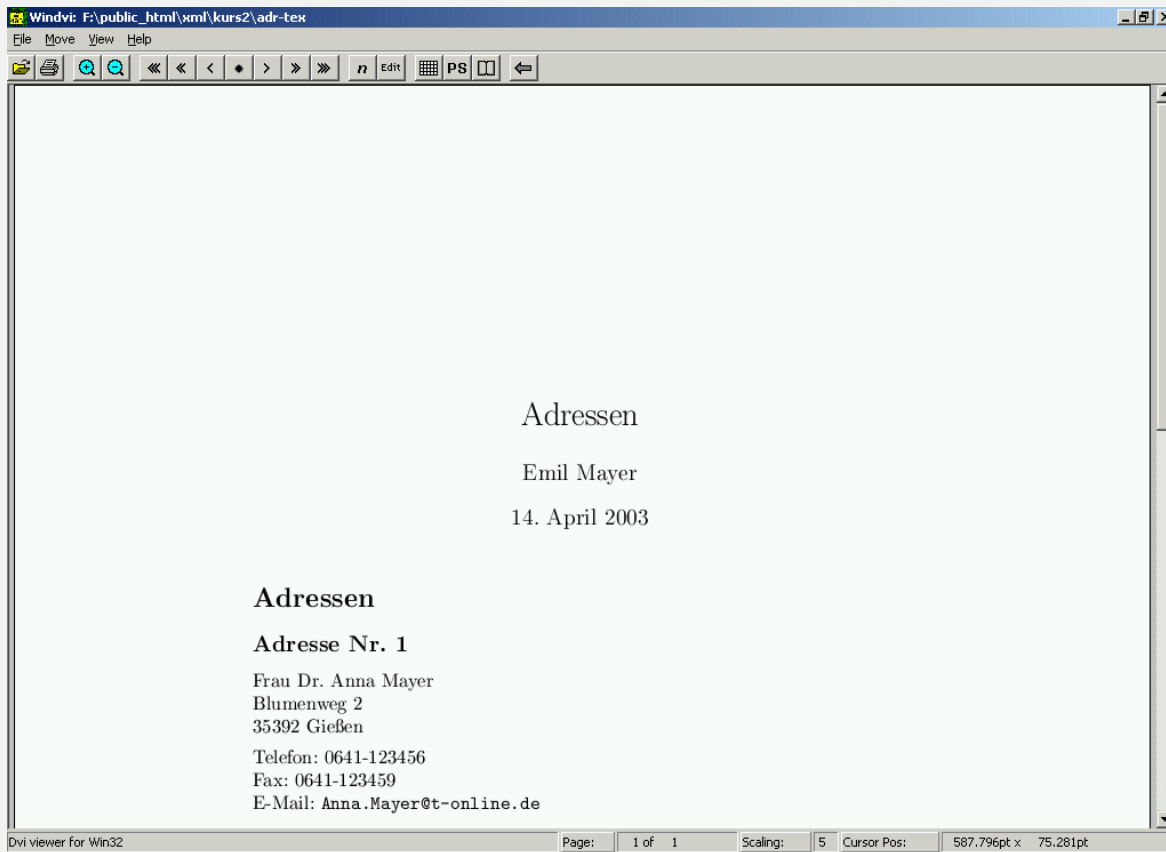
Zurück

Erste

Vorherige

Nächste

Letzte



**Abbildung 1** Darstellung der DVI-Datei adr-tex.dvi im DVI-Previewer Yap

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

xml-adr.awk:

```
1 # xml-adr.awk
2 BEGIN {FS="[\t]"
3     print "<?xml version=\"1.0\" encoding=\"ISO-8859-1\" ?>"
4     print "<!DOCTYPE adressen ["
5     print "<!ELEMENT adressen (adresse*) >"
6     print "<!ELEMENT adresse (anrede?, titel?, "
7     print "        name, strasse?, postfach?, "
8     print "        ort?, telefon?, fax?, email?) >"
9     print "<!ATTLIST adresse nr CDATA #REQUIRED >"
10    print "<!ELEMENT titel (#PCDATA) >"
11    print "<!ELEMENT anrede (#PCDATA) >"
12    print "<!ELEMENT name (vorname?,nachname) >"
13    print "<!ELEMENT vorname (#PCDATA) >"
14    print "<!ELEMENT nachname (#PCDATA) >"
15    print "<!ELEMENT strasse (sname?,hnr?) >"
16    print "<!ELEMENT sname (#PCDATA) >"
17    print "<!ELEMENT hnr (#PCDATA) >"
18    print "<!ELEMENT postfach (#PCDATA) >"
19    print "<!ELEMENT ort (plz?,ortsname?) >"
20    print "<!ELEMENT plz (#PCDATA) >"
21    print "<!ELEMENT ortsname (#PCDATA) >"
22    print "<!ELEMENT telefon (#PCDATA) >"
23    print "<!ELEMENT fax (#PCDATA) >"
24    print "<!ELEMENT email (#PCDATA) >"
25    print "]>"
26    print "<adressen>"}

```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte



```

27 NR==1 {next}
28 NR>=2 {print "<adresse nr=\" NR-1 \">"
29     if($1!="")print "    <anrede>" $1 "</anrede>"
30     if($2!="")print "    <titel>" $2 "</titel>"
31     if($3!="" || $4!="")
32         {print "        <name>"
33             if($3!="")print "            <vorname>" $3 "</vorname>"
34             if($4!="")print "            <nachname>" $4 "</nachname>"
35             print "        </name>"}
36     if($5!="" || $6!="")
37         {print "        <strasse>"
38             if($5!="")print "            <sname>" $5 "</sname>"
39             if($6!="")print "            <hnr>" $6 "</hnr>"
40             print "        </strasse>"}
41     if($7!="")print "    <postfach>" $7 "</postfach>"
42     if($8!="" || $9!="")
43         {print "        <ort>"
44             if($8!="")print "            <plz>" $8 "</plz>"
45             if($9!="")print "            <ortsname>" $9 "</ortsname>"
46             print "        </ort>"}
47     if($10!="")print "    <telefon>" $10 "</telefon>"
48     if($11!="")print "    <fax>" $11 "</fax>"
49     if($12!="")print "    <email>" $12 "</email>"
50     print "    </adresse>"}
51 END {print "</adressen>"}

```

Suchen

Abbruch

adr.xml:

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <!DOCTYPE adressen [
3  <!ELEMENT adressen (adresse*) >
4  <!ELEMENT adresse (anrede?, titel?,
5      name, strasse?, postfach?,
6      ort?, telefon?, fax?, email?) >
7  <!ATTLIST adresse nr CDATA #REQUIRED >
8  <!ELEMENT titel (#PCDATA) >
9  <!ELEMENT anrede (#PCDATA) >
10 <!ELEMENT name (vorname?, nachname) >
11 <!ELEMENT vorname (#PCDATA) >
12 <!ELEMENT nachname (#PCDATA) >
13 <!ELEMENT strasse (sname?, hnr?) >
14 <!ELEMENT sname (#PCDATA) >
15 <!ELEMENT hnr (#PCDATA) >
16 <!ELEMENT postfach (#PCDATA) >
17 <!ELEMENT ort (plz?, ortsname?) >
18 <!ELEMENT plz (#PCDATA) >
19 <!ELEMENT ortsname (#PCDATA) >
20 <!ELEMENT telefon (#PCDATA) >
21 <!ELEMENT fax (#PCDATA) >
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
22 <!ELEMENT email (#PCDATA) >
23 ]>
24 <adressen>
25 <adresse nr="1">
26   <anrede>Frau</anrede>
27   <titel>Dr.</titel>
28   <name>
29     <vorname>Anna</vorname>
30     <nachname>Mayer</nachname>
31   </name>
32   <strasse>
33     <sname>Blumenweg</sname>
34     <hnr>2</hnr>
35   </strasse>
36   <ort>
37     <plz>35392</plz>
38     <ortsname>Gießen</ortsname>
39   </ort>
40   <telefon>0641-123456</telefon>
41   <fax>0641-123459</fax>
42   <email>Anna.Mayer@t-online.de</email>
43 </adresse>
44 <adresse nr="2">
```

Suchen

Abbruch

```
45 <anrede>Herr</anrede>
46 <name>
47   <vorname>Emil</vorname>
48   <nachname>Müller</nachname>
49 </name>
50 <strasse>
51   <sname>Tulpenweg</sname>
52   <hnr>22a</hnr>
53 </strasse>
54 <ort>
55   <plz>35398</plz>
56   <ortsname>Gießen</ortsname>
57 </ort>
58 <telefon>0641-234567</telefon>
59 <email>Emil.Mueller@profanistik.uni-giessen.de</email>
60 </adresse>
61 <adresse nr="3">
62   <anrede>Frau</anrede>
63   <titel>Prof. Dr.</titel>
64   <name>
65     <vorname>Erna</vorname>
66     <nachname>Schneider</nachname>
67   </name>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
68 <strasse>
69   <sname>Bachstraße</sname>
70   <hnr>10</hnr>
71 </strasse>
72 <postfach>12345</postfach>
73 <ort>
74   <plz>12340</plz>
75   <ortsname>Nixdorf</ortsname>
76 </ort>
77 <telefon>09994-1234</telefon>
78 <fax>09994-12349</fax>
79 <email>Erna.Schneider@gmx.de</email>
80 </adresse>
81 </adressen>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

adr2.xml:

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <!-- adr2.xml -->
3  <!DOCTYPE adressen SYSTEM "adressen.dtd" >
4  <adressen>
5  <adresse nr="1">
6    <anrede>Frau</anrede>
7    <titel>Dr.</titel>
8    <name>
9      <vorname>Anna</vorname>
10     <nachname>Mayer</nachname>
11   </name>
12   <strasse>
13     <sname>Blumenweg</sname>
14     <hnr>2</hnr>
15   </strasse>
16   <ort>
17     <plz>35392</plz>
18     <ortsname>Gießen</ortsname>
19   </ort>
20   <telefon>0641-123456</telefon>
21   <fax>0641-123459</fax>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
22  <email>Anna.Mayer@t-online.de</email>
23  </adresse>
24  <adresse nr="2">
25    <anrede>Herr</anrede>
26    <name>
27      <vorname>Emil</vorname>
28      <nachname>Müller</nachname>
29    </name>
30    <strasse>
31      <sname>Tulpenweg</sname>
32      <hnr>22a</hnr>
33    </strasse>
34    <ort>
35      <plz>35398</plz>
36      <ortsname>Gießen</ortsname>
37    </ort>
38    <telefon>0641-234567</telefon>
39    <email>Emil.Mueller@profanistik.uni-giessen.de</email>
40  </adresse>
41  <adresse nr="3">
42    <anrede>Frau</anrede>
43    <titel>Prof. Dr.</titel>
44    <name>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
45     <vorname>Erna</vorname>
46     <nachname>Schneider</nachname>
47     </name>
48     <strasse>
49     <sname>Bachstraße</sname>
50     <hnr>10</hnr>
51     </strasse>
52     <postfach>12345</postfach>
53     <ort>
54     <plz>12340</plz>
55     <ortsname>Nixdorf</ortsname>
56     </ort>
57     <telefon>09994-1234</telefon>
58     <fax>09994-12349</fax>
59     <email>Erna.Schneider@gmx.de</email>
60     </adresse>
61 </adressen>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte



adressen.dtd:

```
1  <!-- adressen.dtd -->
2  <!ELEMENT adressen (adresse*) >
3  <!ELEMENT adresse  (anrede?, titel?, name, strasse?,
4                      postfach?, ort?, telefon?, fax?,
5                      email?) >
6  <!ATTLIST adresse  nr CDATA #REQUIRED >
7  <!ELEMENT titel     (#PCDATA) >
8  <!ELEMENT anrede    (#PCDATA) >
9  <!ELEMENT name      (vorname?, nachname) >
10 <!ELEMENT vorname   (#PCDATA) >
11 <!ELEMENT nachname  (#PCDATA) >
12 <!ELEMENT strasse   (sname?, hnr?) >
13 <!ELEMENT sname     (#PCDATA) >
14 <!ELEMENT hnr       (#PCDATA) >
15 <!ELEMENT postfach  (#PCDATA) >
16 <!ELEMENT ort       (plz?, ortsname?) >
17 <!ELEMENT plz       (#PCDATA) >
18 <!ELEMENT ortsname  (#PCDATA) >
19 <!ELEMENT telefon   (#PCDATA) >
20 <!ELEMENT fax       (#PCDATA) >
21 <!ELEMENT email     (#PCDATA) >
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

pr1.awk:

```
1 # pr1.awk
2 BEGIN
3     {print "<html>"
4     print "<head><title>Adressen</title></head>"
5     print "<body>"
6     print "<h1>Adressen</h1>"
7     print " "}
8 /<adressen>/
9     {print "<table border=1>"
10    print "<tr><!-- Tabellenkopf -->"
11    print "  <td><b>Nr.</b></td>"
12    print "  <td><b>Anrede</b></td>"
13    print "  <td><b>Titel</b></td>"
14    print "  <td><b>Vorname</b></td>"
15    print "  <td><b>Nachname</b></td>"
16    print "  <td><b>PLZ</b></td>"
17    print "  <td><b>Ortsname</b></td>"
18    print "  <td><b>Straßenname</b></td>"
19    print "  <td><b>Hausnummer</b></td>"
20    print "  <td><b>Postfach</b></td>"
21    print "  <td><b>Telefon</b></td>"
22    print "  <td><b>Fax</b></td>"
23    print "  <td><b>EMail</b></td>"
24    print "  </tr>"}
25 /<adresse nr=.*>/
26     {titel=""; anrede=""; vorname=""; nachname="";
27     sname=""; hnr=""; postfach=""; plz="";
28     ortsname=""; telefon=""; fax=""; email=""
29     match($0,/["].*["]/)
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```

27         nr=substr($0,RSTART+1,RLENGTH-2)
28         print "<tr> <!-- Tabelleneintrag " nr " -->"
29 /<titel>/,/<[/]titel>/      {titel=titel $0}
30 /<anrede>/,/<[/]anrede>/    {anrede=anrede $0}
31 /<vorname>/,/<[/]vorname>/  {vorname=vorname $0}
32 /<nachname>/,/<[/]nachname>/ {nachname=nachname $0}
33 /<sname>/,/<[/]sname>/      {sname=sname $0}
34 /<hnr>/,/<[/]hnr>/         {hnr=hnr $0}
35 /<postfach>/,/<[/]postfach>/ {postfach=postfach $0}
36 /<plz>/,/<[/]plz>/         {plz=plz $0}
37 /<ortsname>/,/<[/]ortsname>/ {ortsname=ortsname $0}
38 /<telefon>/,/<[/]telefon>/  {telefon=telefon $0}
39 /<fax>/,/<[/]fax>/         {fax=fax $0}
40 /<email>/,/<[/]email>/     {email=email $0}
41 /<[/]adresse>/            {print " <td>" nr "</td>"
42                             if (anrede=="") anrede="&nbsp;";
43                             print " <td>" anrede "</td>"
44                             if (titel=="") titel="&nbsp;";
45                             print " <td>" titel "</td>"
46                             if (vorname=="") vorname="&nbsp;";
47                             print " <td>" vorname "</td>"
48                             if (nachname=="") nachname="&nbsp;";
49                             print " <td>" nachname "</td>"
50                             if (plz=="") plz="&nbsp;";
51                             print " <td>" plz "</td>"
52                             if (ortsname=="") ortsname="&nbsp;";
53                             print " <td>" ortsname "</td>"
54                             if (sname=="") sname="&nbsp;";

```

Suchen

Abbruch

```

55     print "    <td>" sname "</td>"
56     if (hnr=="") hnr="&nbsp;";
57     print "    <td>" hnr "</td>"
58     if (postfach=="") postfach="&nbsp;";
59     print "    <td>" postfach "</td>"
60     if (telefon=="") telefon="&nbsp;";
61     print "    <td>" telefon "</td>"
62     if (fax=="") fax="&nbsp;";
63     print "    <td>" fax "</td>"
64     if (email=="") email="&nbsp;";
65     print "    <td><tt>" email "</tt></td>"
66     print "  </tr>"
67     titel="";anrede="";vorname="";nachname="";
68     sname="";hnr="";postfach="";plz="";
69     ortsname="";telefon="";fax="";email=""}
70 /<[\\/]adressen>/ {print "</table>"}
71 END {print "</body>"
72      print "</html>"}

```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

adr.html:

```
1  <html>
2  <head><title>Adressen</title></head>
3  <body>
4  <h1>Adressen</h1>

5  <table border=1>
6  <tr><!-- Tabellenkopf -->
7      <td><b>Nr.</b></td>
8      <td><b>Anrede</b></td>
9      <td><b>Titel</b></td>
10     <td><b>Vorname</b></td>
11     <td><b>Nachname</b></td>
12     <td><b>PLZ</b></td>
13     <td><b>Ortsname</b></td>
14     <td><b>Straßenname</b></td>
15     <td><b>Hausnummer</b></td>
16     <td><b>Postfach</b></td>
17     <td><b>Telefon</b></td>
18     <td><b>Fax</b></td>
19     <td><b>EMail</b></td>
20 </tr>
21 <tr> <!-- Tabelleneintrag 1 -->
22     <td>1</td>
23     <td><anrede>Frau</anrede></td>
24     <td><titel>Dr.</titel></td>
25     <td><vorname>Anna</vorname></td>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```

26 <td><nachname>Mayer</nachname></td>
27 <td><plz>35392</plz></td>
28 <td><ortsname>Gießen</ortsname></td>
29 <td><sname>Blumenweg</sname></td>
30 <td><hnr>2</hnr></td>
31 <td>&nbsp;</td>
32 <td><telefon>0641-123456</telefon></td>
33 <td><fax>0641-123459</fax></td>
34 <td><tt><email>Anna.Mayer@t-online.de</email></tt></td>
35 </tr>
36 <tr> <!-- Tabelleneintrag 2 -->
37 <td>2</td>
38 <td><anrede>Herr</anrede></td>
39 <td>&nbsp;</td>
40 <td><vorname>Emil</vorname></td>
41 <td><nachname>Müller</nachname></td>
42 <td><plz>35398</plz></td>
43 <td><ortsname>Gießen</ortsname></td>
44 <td><sname>Tulpenweg</sname></td>
45 <td><hnr>22a</hnr></td>
46 <td>&nbsp;</td>
47 <td><telefon>0641-234567</telefon></td>
48 <td>&nbsp;</td>
49 <td><tt><email>Emil.Mueller@profanistik.uni-giessen.de</email></tt></td>
50 </tr>
51 <tr> <!-- Tabelleneintrag 3 -->
52 <td>3</td>
53 <td><anrede>Frau</anrede></td>

```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
54 <td><titel>Prof. Dr.</titel></td>
55 <td><vorname>Erna</vorname></td>
56 <td><nachname>Schneider</nachname></td>
57 <td><plz>12340</plz></td>
58 <td><ortsname>Nixdorf</ortsname></td>
59 <td><sname>Bachstraße</sname></td>
60 <td><hnr>10</hnr></td>
61 <td><postfach>12345</postfach></td>
62 <td><telefon>09994-1234</telefon></td>
63 <td><fax>09994-12349</fax></td>
64 <td><tt><email>Erna.Schneider@gmx.de</email></tt></td>
65 </tr>
66 </table>
67 </body>
68 </html>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

adr.css:

```
1  /* adr.css */
2  postfach:before {content:"Postfach: "; }
3  telefon:before  {content:"Tel.: "; }
4  fax:before      {content:"Fax: "; }
5  email:before    {content:"E-Mail: "; }
6  adressen:before {content:"Adressen "; }
7  adressen:after  {content:"-----"; }
8  adressen        {color:#CC0000;
9                  font-size:130%; }
10 adresse        {display:block;
11                 margin-top:1.0ex;
12                 font-family:sans-serif; }
13 ort             {display:block;
14                 font-weight:bold;
15                 margin-bottom:0.5ex; }
16 strasse         {display:block; }
17 telefon         {display:block; }
18 fax             {display:block; }
19 email           {display:block;
20                 font-family:monospace; }
21 postfach        {display:block; }
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte



adr3.xml:

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <!-- adr3.xml -->
3  <?xml-stylesheet href="adr.css" type="text/css" ?>
4  <!DOCTYPE adressen SYSTEM "adressen.dtd" >
5  <adressen>
6  <adresse nr="1">
7    <anrede>Frau</anrede>
8    <titel>Dr.</titel>
9    <name>
10      <vorname>Anna</vorname>
11      <nachname>Mayer</nachname>
12    </name>
13    <strasse>
14      <sname>Blumenweg</sname>
15      <hnr>2</hnr>
16    </strasse>
17    <ort>
18      <plz>35392</plz>
19      <ortsname>Gießen</ortsname>
20    </ort>
21    <telefon>0641-123456</telefon>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
22 <fax>0641-123459</fax>
23 <email>Anna.Mayer@t-online.de</email>
24 </adresse>
25 <adresse nr="2">
26 <anrede>Herr</anrede>
27 <name>
28 <vorname>Emil</vorname>
29 <nachname>Müller</nachname>
30 </name>
31 <strasse>
32 <sname>Tulpenweg</sname>
33 <hnr>22a</hnr>
34 </strasse>
35 <ort>
36 <plz>35398</plz>
37 <ortsname>Gießen</ortsname>
38 </ort>
39 <telefon>0641-234567</telefon>
40 <email>Emil.Mueller@profanistik.uni-giessen.de</email>
41 </adresse>
42 <adresse nr="3">
43 <anrede>Frau</anrede>
44 <titel>Prof. Dr.</titel>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
45 <name>
46   <vorname>Erna</vorname>
47   <nachname>Schneider</nachname>
48 </name>
49 <strasse>
50   <sname>Bachstraße</sname>
51   <hnr>10</hnr>
52 </strasse>
53 <postfach>12345</postfach>
54 <ort>
55   <plz>12340</plz>
56   <ortsname>Nixdorf</ortsname>
57 </ort>
58 <telefon>09994-1234</telefon>
59 <fax>09994-12349</fax>
60 <email>Erna.Schneider@gmx.de</email>
61 </adresse>
62 </adressen>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

adr5.xml:

```
1  <?xml version="1.0" encoding="ISO-8859-1" ?>
2  <!-- adr5.xml -->
3  <?xml-stylesheet href="adr.xsl" type="text/xsl" ?>
4  <!DOCTYPE adressen SYSTEM "adressen.dtd" >
5  <adressen>
6  <adresse nr="1">
7    <anrede>Frau</anrede>
8    <titel>Dr.</titel>
9    <name>
10      <vorname>Anna</vorname>
11      <nachname>Mayer</nachname>
12    </name>
13    <strasse>
14      <sname>Blumenweg</sname>
15      <hnr>2</hnr>
16    </strasse>
17    <ort>
18      <plz>35392</plz>
19      <ortsname>Gießen</ortsname>
20    </ort>
21    <telefon>0641-123456</telefon>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
22 <fax>0641-123459</fax>
23 <email>Anna.Mayer@t-online.de</email>
24 </adresse>
25 <adresse nr="2">
26 <anrede>Herr</anrede>
27 <name>
28 <vorname>Emil</vorname>
29 <nachname>Müller</nachname>
30 </name>
31 <strasse>
32 <sname>Tulpenweg</sname>
33 <hnr>22a</hnr>
34 </strasse>
35 <ort>
36 <plz>35398</plz>
37 <ortsname>Gießen</ortsname>
38 </ort>
39 <telefon>0641-234567</telefon>
40 <email>Emil.Mueller@profanistik.uni-giessen.de</email>
41 </adresse>
42 <adresse nr="3">
43 <anrede>Frau</anrede>
44 <titel>Prof. Dr.</titel>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
45 <name>
46   <vorname>Erna</vorname>
47   <nachname>Schneider</nachname>
48 </name>
49 <strasse>
50   <sname>Bachstraße</sname>
51   <hnr>10</hnr>
52 </strasse>
53 <postfach>12345</postfach>
54 <ort>
55   <plz>12340</plz>
56   <ortsname>Nixdorf</ortsname>
57 </ort>
58 <telefon>09994-1234</telefon>
59 <fax>09994-12349</fax>
60 <email>Erna.Schneider@gmx.de</email>
61 </adresse>
62 </adressen>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

adr.xsl:

```
1  <?xml version='1.0'?>
2  <!-- adr.xsl -->
3  <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
4    <xsl:template match="/">
5      <html>
6        <head><title>Adressen</title></head>
7
8        <body>"
9
10       <xsl:for-each select="adressen/adresse">
11         <p>
12           <xsl:value-of select="anrede" />
13           <xsl:value-of select="titel"/>
14           <xsl:value-of select="name/vorname"/>
15           <xsl:value-of select="name/nachname"/><br />
16           <xsl:value-of select="strasse/sname"/>
17           <xsl:value-of select="strasse/hnr"/><br />
18           <xsl:if test="postfach">
19             Postfach: <xsl:value-of select="postfach"/><br />
20           </xsl:if>
21         </p>
22       </xsl:for-each>
23     </template>
24   </stylesheet>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```
20      <xsl:value-of select="ort/plz"/>
21      <xsl:value-of select="ort/ortsname"/><br />
22      <xsl:if test="telefon">
23          Telefon: <xsl:value-of select="telefon"/><br />
24      </xsl:if>
25      <xsl:if test="fax">
26          Fax: <xsl:value-of select="fax"/><br />
27      </xsl:if>
28      E-Mail: <tt><xsl:value-of select="email"/></tt>
29      </p>
30  </xsl:for-each>
31
32  </body>
33  </html>
34  </xsl:template>
35  </xsl:stylesheet>
```

Suchen

Abbruch



adr2.xsl:

```
1  <?xml version='1.0' encoding="ISO-8859-1" ?>
2  <!-- adr2.xsl -->
3  <xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
4    <xsl:template match="/">
5      <html>
6        <head><title>Adressen</title></head>
7
8        <body>
9          <h1>Adressen</h1>
10
11         <table border="1">
12           <tr>
13             <td><b>Nr.</b></td>
14             <td><b>Anrede</b></td>
15             <td><b>Titel</b></td>
16             <td><b>Vorname</b></td>
17             <td><b>Name</b></td>
18             <td><b>Straße</b></td>
19             <td><b>Nr.</b></td>
20             <td><b>Postfach</b></td>
21             <td><b>PLZ</b></td>
22             <td><b>Ort</b></td>
23             <td><b>Telefon</b></td>
24             <td><b>Fax</b></td>
25             <td><b>E-Mail</b></td>
26           </tr>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```

25 <xsl:for-each select="adressen/adresse" order-by="ort/plz">
26   <tr>
27     <td><xsl:value-of select="@nr" /></td>
28     <td><xsl:value-of select="anrede" /></td>
29     <td><xsl:value-of select="titel"/></td>
30     <td><xsl:value-of select="name/vorname"/></td>
31     <td><xsl:value-of select="name/nachname"/></td>
32     <td><xsl:value-of select="strasse/sname"/></td>
33     <td><xsl:value-of select="strasse/hnr"/></td>
34     <td><xsl:value-of select="postfach"/></td>
35     <td><xsl:value-of select="ort/plz"/></td>
36     <td><xsl:value-of select="ort/ortsname"/></td>
37     <td><xsl:value-of select="telefon"/></td>
38     <td><xsl:value-of select="fax"/></td>
39     <td><TT><xsl:value-of select="email"/></TT></td>
40   </tr>
41 </xsl:for-each>
42 </table>
43 </body>
44 </html>
45 </xsl:template>
46 </xsl:stylesheet>

```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

## adr-tex.xsl:

```
1  <?xml version='1.0' encoding="ISO-8859-1" ?>
2  <!-- adr-tex.xsl -->
3  <!-- XSL-Datei (passend zu adr5.xml -->
4  <!-- kann mit SAXON verarbeitet werden -->
5  <!-- 28.2.2001, GP (HRZ Gießen) -->
6  <xsl:stylesheet version="1.0"
7    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
8    <xsl:output method="text" encoding="ISO-8859-1" />
9    <xsl:template match="/">
10      \documentclass[10pt,a4paper]{article}
11      \usepackage{adr-tex}
12
13      \title{Adressen}
14      \author{Emil Mayer}
15      \date{\today}
16
17      \begin{document}
18      \maketitle
19      \section*{Adressen}
20
21      <xsl:for-each select="adressen/adresse">
22        \begin{adresse}{<xsl:value-of select="@nr" />}<xsl:text> </xsl:text>
23        \anrede{<xsl:value-of select="anrede" />}<xsl:text> </xsl:text>
24        <xsl:if test="titel">
25          \titel{<xsl:value-of select="titel"/>}<xsl:text> </xsl:text>
26        </xsl:if>
27      \end{adresse}
28    </xsl:for-each>
29  </xsl:template>
30 </xsl:stylesheet>
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

```

24 \vorname{<xsl:value-of select="name/vorname"/><xsl:text> </xsl:text>
25 \nachname{<xsl:value-of select="name/nachname"/><xsl:text> </xsl:text>
26 \sname{<xsl:value-of select="strasse/sname"/><xsl:text> </xsl:text>
27 \hnr{<xsl:value-of select="strasse/hnr"/><xsl:text> </xsl:text>
28 <xsl:if test="postfach">
29   \postfach{<xsl:value-of select="postfach"/><xsl:text> </xsl:text>
30   </xsl:if>
31 \plz{<xsl:value-of select="ort/plz"/><xsl:text> </xsl:text>
32 \ortsname{<xsl:value-of select="ort/ortsname"/><xsl:text> </xsl:text>
33 <xsl:if test="telefon">
34   \telefon{<xsl:value-of select="telefon"/><xsl:text> </xsl:text>
35   </xsl:if>
36 <xsl:if test="fax">
37   \fax{<xsl:value-of select="fax"/><xsl:text> </xsl:text>
38   </xsl:if>
39 <xsl:if test="email">
40   \email{<xsl:value-of select="email"/><xsl:text> </xsl:text>
41   </xsl:if>
42 \end{adresse}
43 </xsl:for-each>

44 \end{document}
45 </xsl:template>
46 </xsl:stylesheet>

```

Suchen

Abbruch

adr-tex.tex:

```
1      \documentclass[10pt,a4paper]{article}
2      \usepackage{adr-tex}

3      \title{Adressen}
4      \author{Emil Mayer}
5      \date{\today}

6      \begin{document}
7      \maketitle
8      \section*{Adressen}

9      \begin{adresse}{1}
10     \anrede{Frau}
11     \titel{Dr.}
12     \vorname{Anna}
13     \nachname{Mayer}
14     \sname{Blumenweg}
15     \hnr{2}
16     \plz{35392}
17     \ortsname{Gießen}
18     \telefon{0641-123456}
```

[Suchen](#)

[Abbruch](#)

[Zurück](#)

[Erste](#)

[Vorherige](#)

[Nächste](#)

[Letzte](#)

```
19      \fax{0641-123459}  
20      \email{Anna.Mayer@t-online.de}  
21  \end{adresse}  
  
22  \begin{adresse}{2}  
23  \anrede{Herr}  
24  \vorname{Emil}  
25  \nachname{Müller}  
26  \sname{Tulpenweg}  
27  \hnr{22a}  
28  \plz{35398}  
29  \ortsname{Gießen}  
30      \telefon{0641-234567}  
31      \email{Emil.Mueller@profanistik.uni-giessen.de}  
32  \end{adresse}  
  
33  \begin{adresse}{3}  
34  \anrede{Frau}  
35      \titel{Prof. Dr.}  
36  \vorname{Erna}  
37  \nachname{Schneider}  
38  \sname{Bachstraße}  
39  \hnr{10}
```

Suchen

Abbruch

```
40      \postfach{12345}  
41      \plz{12340}  
42      \ortsname{Nixdorf}  
43      \telefon{09994-1234}  
44      \fax{09994-12349}  
45      \email{Erna.Schneider@gmx.de}  
46      \end{adresse}  
  
47      \end{document}
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte

## adr-tex.sty:

```
1 % adr-tex.sty
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{adr-tex}[2001/02/28 adr-tex.sty]
4 \immediate\write16{This is adr-tex.sty,%
5     Version 1.00 <2001/02/28>}

6 \RequirePackage[latin1]{inputenc}
7 \RequirePackage{ngerman}

8 \newenvironment{adresse}[1]{\subsection*{Adresse~Nr.~\#1}}{}
9 \newcommand{\anrede}[1]{\#1}
10 \newcommand{\titel}[1]{\#1}
11 \newcommand{\vorname}[1]{\#1}
12 \newcommand{\nachname}[1]{\#1\\}
13 \newcommand{\sname}[1]{\#1}
14 \newcommand{\hnr}[1]{\#1\\}
15 \newcommand{\postfach}[1]{Postfach:~\#1\\}
16 \newcommand{\plz}[1]{\#1}
17 \newcommand{\ortsname}[1]{\#1\\[1.0ex]}
18 \newcommand{\telefon}[1]{Telefon:~\#1\\}
19 \newcommand{\fax}[1]{Fax:~\#1\\}
20 \newcommand{\email}[1]{E-Mail:~\texttt{\#1}}
21 \endinput
```

Suchen

Abbruch

Zurück

Erste

Vorherige

Nächste

Letzte



$$x^2 + 4x + 4 = 0$$

$$a^2 = b^2 + c^2$$

# Adressen

## Adresse Nr. 1

Frau Dr. Anna Mayer  
Blumenweg 2  
35392 Gießen  
Telefon: 0641-123456  
Fax: 0641-123459  
E-Mail: `Anna.Mayer@t-online.de`

## Adresse Nr. 2

Herr Emil Müller  
Tulpenweg 22a  
35398 Gießen  
Telefon: 0641-234567  
E-Mail: `Emil.Mueller@profanistik.uni-giessen.de`

## Adresse Nr. 3

Frau Prof. Dr. Erna Schneider  
Bachstraße 10  
Postfach: 12345  
12340 Nixdorf  
Telefon: 09994-1234  
Fax: 09994-12349  
E-Mail: `Erna.Schneider@gmx.de`

# Adressen

Emil Mayer

4. Februar 2004

## Adressen

### Adresse~Nr.~1

Frau Dr. Anna Mayer  
Blumenweg 2  
35392 Gießen

Telefon:~0641-123456

Fax:~0641-123459

E-Mail:~Anna.Mayer@t-online.de

### Adresse~Nr.~2

Herr Emil Müller  
Tulpenweg 22a  
35398 Gießen

Telefon:~0641-234567

E-Mail:~Emil.Mueller@profanistik.uni-giessen.de

### Adresse~Nr.~3

Frau Prof. Dr. Erna Schneider  
Bachstraße 10  
Postfach:~12345  
12340 Nixdorf

Telefon:~09994-1234

Fax:~09994-12349

E-Mail:~Erna.Schneider@gmx.de

# Unser Kapitel

Emil Mayer

4. Februar 2004

## **1 Titel des ersten Kapitels**

Ein bisschen Text ...

    Noch ein bisschen Text ...

## **2 Titel des zweiten Kapitels**

Ein bisschen anderer Text ...

    Noch ein bisschen anderer Text ...

# Unser Kapitel

Emil Mayer

4. Februar 2004

## **1 Titel des ersten Kapitels**

Ein bisschen Text ...

    Noch ein bisschen Text ...

## **2 Titel des zweiten Kapitels**

Ein bisschen anderer Text ...

    Noch ein bisschen anderer Text ...