Die hier verkürzt vorgestellten XSLT-Deklarationen und -Instruktionen `xsl:stylesheet`, `xsl:template`, `xsl:apply-templates`, `xsl:value-of`, `xsl:text`, `xsl:output`, `xsl:preserve-space` und `xsl:strip-space` werden in einigen Fällen im weiteren Verlauf des Einführungsteils vertieft. Eine ausführliche Erörterung aller XSLT-Elemente und ihrer Attribute befindet sich jeweils gesammelt in Kapitel 6, *Referenz XSLT-Elemente*.

1.19.1 Testfragen

Folgende Fragen sollten Sie ohne große Schwierigkeiten beantworten können. Eine oder mehrere Antworten sind jeweils richtig. Die erläuterten Auflösungen mit zusätzlichen Hintergrundinformationen finden Sie gesammelt in Anhang A.

- **Frage 1:** Welcher Knotentyp ist nicht »Child« seines Elternknotens?
  a) Der Textknoten
  b) Der Attributknoten
  c) Der Kommentarknoten
  d) Der Processing-Instructions-Knoten
  e) Der Namensraumknoten
- **Frage 2:** Was ist der »value« eines Elementknotens?
  a) Text- und Elementinhalt
  b) Attributwerte und Textinhalte
  c) Eigene Textinhalte und Textinhalte von Child-Elementen
  d) Nur die unmittelbaren Child-Textknoten
- **Frage 3:** Der Document Node ist ...
  a) ... das Wurzelelement des verarbeiteten XML-Dokuments.
  b) ... das Wurzelelement des XSLT-Stylesheets.
  c) ... ein virtueller Knoten, der das verarbeitete XML-Dokument enthält.
- **Frage 4:** Warum ist der XSL-Namensraum-URI bei `xsl:stylesheet` wichtig?
  a) Der Prozessor holt sich Anweisungen unter dieser Adresse.
  b) Damit erst werden XSLT-Elemente als Befehle kenntlich.
  c) Ganz unwichtig. Hauptsache das Präfix lautet `xsl:`
- **Frage 5:** Welche XSLT-Instruktion stellt eine Nodesequenz zusammen?
  a) `xsl:stylesheet`
  b) `xsl:template`
  c) `xsl:apply-templates`
  d) `xsl:value-of`

- **Frage 6:** Womit wird der XSLT-Prozess eingeleitet?
  a) Mit einem Template mit Match auf den Document Node
  b) Mit dem ersten Template in Dokumentreihenfolge
  c) Mit dem Template mit höchster Priorität

1.19.2 Aufgaben

Verwenden Sie zur Bearbeitung der Aufgaben AltovaXML oder Saxon 9 als XSLT-Prozessor (alternativ dessen GUI Kernow). Die erforderlichen Installationsdateien des hierfür vorausgesetzten JDK 1.5 befinden sich auf der Begleit-CD. Installation und Bedienung der Prozessoren sind in Kapitel 7, *XSLT-Prozessoren für XSLT 2.0*, beschrieben.

Rohmaterial für die folgenden Aufgaben ist ein einfaches XML-Dokument, das – ähnlich einer E-Mail – eine kurze Textmitteilung enthält:[26]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<memo>
<an>Peter Panter</an>
<von>Theobald Tiger</von>
<betreff>Neues Buch von Tucholsky</betreff>
<text>Hallo Peter! Hast Du schon das neue Buch von Tucholsky
gelesen? Gruss, Theo!</text>
</memo>
```

Listing 1.42 aufgaben/kap01/memo1.xml

Zur Verarbeitung ist ein unfertiges Stylesheet-Dokument vorhanden, das ergänzt werden soll:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="2.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template ...>
<html>
<body>
<h1>Memo:</h1>

</body>
</html>
```

[26] Peter Panter und Theobald Tiger waren von Kurt Tucholsky verwendete Pseudonyme, unter denen er als Kritiker auftrat.

```
</xsl:template>
</xsl:stylesheet>
```

Listing 1.43 aufgaben/kap01/memo.xsl

- **Aufgabe 1a:** Stringwert ausgeben
  Fügen Sie der Template-Regel ein geeignetes `match`-Attribut hinzu. Ergänzen Sie das Stylesheet so, dass der Stringwert des Quelldokuments nach der Überschrift in einem `<p>`-Container ausgegeben wird. Verwenden Sie dazu die Instruktion `xsl:value-of`.
- **Aufgabe 2a:** Stringwerte der Einzelelemente ausgeben
  Erweitern Sie das Stylesheet aus 1) so, dass im Ergebnisdokument die Inhalte der einzelnen Elemente jeweils in eigenen `<p>`-Containern stehen. Denken Sie daran, dass XPath-Ausdrücke einen Pfad vom Current Node zum Zielknoten darstellen!
- **Aufgabe 2b:** Reihenfolge kontrollieren
  Schreiben Sie eine Variation von 2), bei der der Inhalt von `<von>` vor dem Inhalt von `<an>` ausgegeben wird. Geben Sie zusätzlich (fett markiert) die Zeichenketten **Von:** und **An:** und **Betreff:** vor den entsprechenden Inhalten aus.
- **Aufgabe 2c:** Reihenfolge kontrollieren, alternative Version
  Lösen Sie die Aufgabe 2a) auf Grundlage von 1a), also mit `xsl:apply-templates`.
- **Aufgabe 3:** Mehrfachverwendung von Inhalten
  Eine HTML-Seite sollte einen Dokumenttitel besitzen. Im Titel sollte stehen »Memo von (Absendername) an (Empfängername)«. Kein Problem?
- **Aufgabe 4a:** Mehrere Template-Regeln
  Erweitern Sie das Stylesheet durch mehrere Template-Regeln, die durch `xsl:apply-templates` angesprochen werden. Das `match`-Attribut muss gemäß dem zu verarbeitenden Element gesetzt werden. Belassen Sie die Ausgabereihenfolge wie vom Dokument vorgegeben.
- **Aufgabe 4b:** Mehrere Template-Regeln, Reihenfolge kontrollieren
  Versuchen Sie, das gleiche Ergebnis wie in 2a) zu erzielen.
- **Aufgabe 5:** Attribute auslesen
  Verwenden Sie folgendes Quelldokument:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<buch titel="XML - Das verflixte Attribut"
autor="Mike Müller"
übersetzer="Peter Panter"
verlag="Galileo Computing"
```



```
isbn="0 815 12345 0"
ersch-jahr="2010"/>
```

Listing 1.44 aufgaben/kap01/buch-attrib.xml

Erstellen Sie ein Stylesheet, das ein Template mit Match auf den Document Node enthält. Schreiben Sie in dieses das HTML-Grundgerüst. Bauen Sie `xsl:value-of`-Anweisungen ein, die die Attribute auslesen. Versuchen Sie eine Formatierung in etwa folgender Form zu erreichen:

Mike Miller: »XML – Das verflixte Attribut«
in der Übersetzung von *Peter Panter*
Galileo Computing, 2010

- **Aufgabe 6a:** Verwenden von `xsl:output`
  Geben Sie anstelle des in Aufgabe 5 generierten HTML-Dokuments ein XHTML-Dokument aus. Verwenden Sie `xsl:output` mit entsprechenden `encoding`- und `indent`-Attributen. Setzen Sie den XHTML-Namensraum einmal im Stylesheet-Element und einmal alternativ im `<html>`-Wurzelelement. Vergleichen Sie die Ergebnisse. Lassen Sie versuchsweise das `indent`-Attribut weg oder setzen Sie es auf `indent="no"`.
- **Aufgabe 6b:** Verwenden von `xsl:output`
  Verwenden Sie versuchsweise `method="xml"`. Wie unterscheidet sich das Ergebnis von der vorigen Ausgabe? Probieren Sie auch `method="text"`.