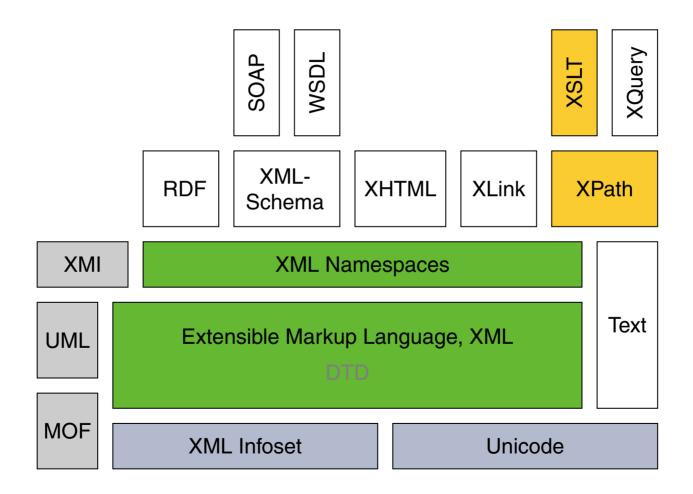
Kapitel WT:III (Fortsetzung)

III. Dokumentsprachen

- □ Auszeichnungssprachen
- □ HTML
- Cascading Stylesheets CSS
- □ XML-Grundlagen
- □ XML-Schema
- □ Die XSL-Familie
- □ APIs für XML-Dokumente

WT:III-276 Dokumentsprachen © STEIN 2022

Einordnung [Jeckle 2004]



WT:III-277 Dokumentsprachen ©STEIN 2022

Die XSL-Familie [W3C xsl home, reports]

Historie: zentrale XML-Spezifikationen

- 2006 Extensible Markup Language (XML) 1.1. Recommendation. [W3C REC, status]
- 2004 XML Schema Part 0: Primer. Recommendation. [W3C REC, status]
- 2012 XML Schema (XSD) 1.1 Part 1: Structures. [W3C REC]
- 2012 XML Schema (XSD) 1.1 Part 2: Datatypes. [W3C REC]
- 2021 XSL Transformations (XSLT) 2.0. Recommendation. [W3C REC, status]
- 2017 XML Path Language (XPath) 3.1. Recommendation. [W3C REC, status]
- 2017 XML Query Language (XQuery) 3.1. Recommendation. [W3C REC, status]
- 2012 XSL Formatting Objects (XSL-FO) 2.0. Working Draft. [W3C WD, Wiki]

WT:III-278 Dokumentsprachen ©STEIN 2022

Bemerkungen:

- "[The extensible stylesheet language family] XSL is a family of recommendations for defining XML document transformation and presentation. It consists of three parts:" XSLT, XPath, XSL-FO. [W3C]
- □ CSS versus XSL. Why two Style Sheet languages? [W3C 1, 2]
- □ Die Formatierungsmöglichkeiten von XSL-FO orientier(t)en sich an den Anforderungen von Print-Medien und ziel(t)en auf eine Ablösung von PDF. Schwerpunkte der Standards: CSS für HTML und XSL-FO für Print.

Die Entwicklung des XSL-FO-Standards wurde 2013 gestoppt und wird durch die Entwicklung von CSS3-page fortgesetzt. [stackoverflow, readwritecode]

Schritte eines XSL-Verarbeitungsprozesses [W3C]:



WT:III-279 Dokumentsprachen © STEIN 2022

XForm

Verwendung von XPath

XSLT Finden und Auswählen von Elementen im Eingabedokument, die in das Ausgabedokument kopiert/transformiert werden.

XQuery Finden und Auswählen von Elementen.

XPointer Identifikation einer Stelle im XML-Dokument,

auf die ein XLink verweist.

XML-DOM-API XPath-Interface zum Zugriff auf den DOM.

XML-Schema Formulierung von Constraints hinsichtlich der Eindeutigkeit oder der Identität von Elementen.

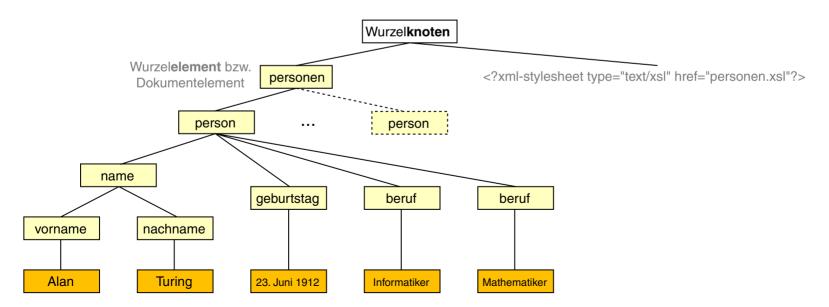
Bindung von Formularsteuerungen an Instanzdaten;

Formulierung von Werte-Constraints und Berechnungen.

WT:III-280 Dokumentsprachen ©STEIN 2022

XML-Knotentypen unter dem XPath-Modell

- 1. Wurzelknoten
- Elementknoten
- 3. Textknoten
- 4. Attributknoten
- 5. Kommentarknoten
- 6. Verarbeitungsanweisungsknoten
- 7. Namensraumknoten



WT:III-281 Dokumentsprachen ©STEIN 2022

Bemerkungen:

- Der Wurzelknoten eines XML-Dokuments ist nicht identisch mit dem Wurzelelement: Der Wurzelknoten entspricht dem Document Information Item des XML Information Sets. Das Wurzelelement hingegen ist das erste benannte Element des Dokuments und wird durch ein Element Information Item dargestellt.
- XPath dient zur Navigation in Dokumenten und der Auswahl von Dokumentbestandteilen;
 XPath ist keine Datenmanipulationssprache.
- XPath-Ausdrücke können zu einzelnen Knoten (XML-Element, XML-Attribut), zu Knotenmengen, zu Zeichenketten, zu Zahlen und zu Bool'schen Werten evaluieren. XPath stellt deshalb Funktionen zum Zugriff auf Knotenmengen und zur Manipulation verschiedener Datentypen zur Verfügung.
- □ Wiederholung. Das W3C hat mittlerweile drei Datenmodelle für XML-Dokumente definiert: XML Information Set, XPath, Document Object Model (DOM). Das XPath-Datenmodell basiert auf einer Baumstruktur, die bei der Abfrage eines XML-Dokuments durchlaufen wird und ist dem XML Information Set ähnlich; DOM ist der Vorläufer beider Datenmodelle. DOM und das XPath-Datenmodell können als Interpretationen des XML Information Sets betrachtet werden. [MSDN]

WT:III-282 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade

- □ Ein Lokalisierungspfad spezifiziert eine eventuell leere Menge von Knoten in einem XML-Dokument.
- Ein Lokalisierungspfad setzt sich aus aufeinander folgenden Lokalisierungsschritten (Location steps) zusammen.
- Jeder Lokalisierungsschritt wird relativ zu einem bestimmten Knoten des XML-Dokuments ausgewertet, der dann als aktueller Knoten (*Current node*) oder Kontextknoten bezeichnet wird.

WT:III-283 Dokumentsprachen © STEIN 2022

XPath-Lokalisierungspfade

- □ Ein Lokalisierungspfad spezifiziert eine eventuell leere Menge von Knoten in einem XML-Dokument.
- Ein Lokalisierungspfad setzt sich aus aufeinander folgenden Lokalisierungsschritten (Location steps) zusammen.
- Jeder Lokalisierungsschritt wird relativ zu einem bestimmten Knoten des XML-Dokuments ausgewertet, der dann als aktueller Knoten (*Current node*) oder Kontextknoten bezeichnet wird.

□ Lokalisierungsschritte werden durch Schrägstriche (*Slashes*) getrennt:

```
... / Schritt_i / Schritt_i+1 / ...
```

Beginnt ein Lokalisierungspfad mit einem Schrägstrich, bezeichnet dieser den Wurzelknoten. Der Wurzelknoten ist dann Kontextknoten zum ersten Lokalisierungsschritt:

WT:III-284 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

(a) /personen/person

WT:III-285 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
```

Beispiele für Lokalisierungspfade:

(a) /personen/person

WT:III-286 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

- (a) /personen/person
- (b) /personen/person[1]/name/vorname

WT:III-287 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
    <vorname>Alan
```

Beispiele für Lokalisierungspfade:

- (a) /personen/person
- (b) /personen/person[1]/name/vorname

WT:III-288 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

- (a) /personen/person
- (b) /personen/person[1]/name/vorname
- (C) /personen/person[1]/beruf

WT:III-289 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
```

Beispiele für Lokalisierungspfade:

- (a) /personen/person
- (b) /personen/person[1]/name/vorname
- (C) /personen/person[1]/beruf

WT:III-290 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

Allgemeine Form eines Lokalisierungsschritts:

```
... / Achse::Knotentest[Prädikat] / ...
```

1. Achse. Spezifiziert eine Knotenmenge relativ zum Kontextknoten. Es werden 13 Achsen unterschieden, child:: ist die Defaultachse.

WT:III-291 Dokumentsprachen © STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

Allgemeine Form eines Lokalisierungsschritts:

```
... / Achse::Knotentest[Prädikat] / ...
```

- 1. Achse. Spezifiziert eine Knotenmenge relativ zum Kontextknoten. Es werden 13 Achsen unterschieden, child:: ist die Defaultachse.
- 2. Knotentest. Filtert die durch eine Achse (1.) spezifizierte Knotenmenge weiter. Hierzu gibt es für jeden Knotentyp ein Testschema.

Beispiele:

- □ ein qualifizierender Name (wie "Person")

 Test auf Knoten mit diesem Namen

WT:III-292 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

Allgemeine Form eines Lokalisierungsschritts:

```
... / Achse::Knotentest[Prädikat] / ...
```

- 1. Achse. Spezifiziert eine Knotenmenge relativ zum Kontextknoten. Es werden 13 Achsen unterschieden, child:: ist die Defaultachse.
- 2. Knotentest. Filtert die durch eine Achse (1.) spezifizierte Knotenmenge weiter. Hierzu gibt es für jeden Knotentyp ein Testschema.

Beispiele:

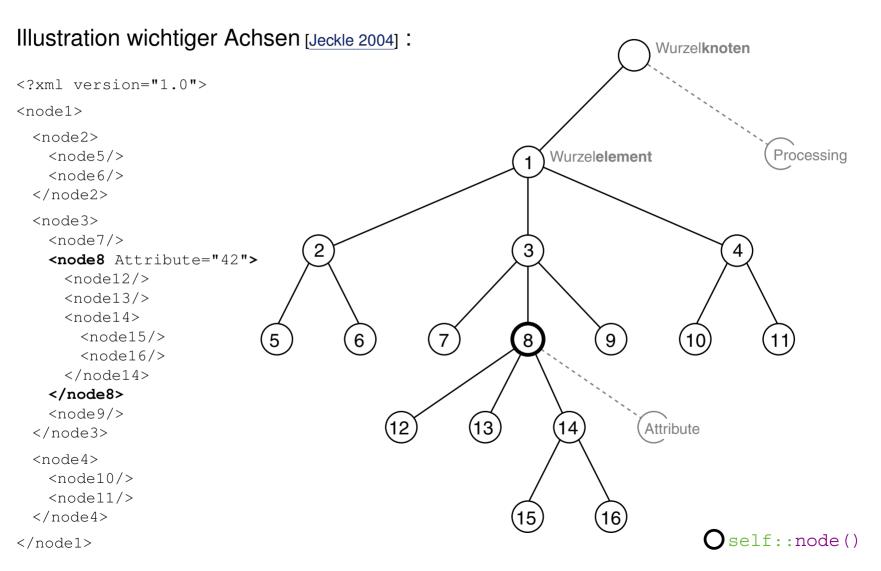
- □ ein qualifizierender Name (wie "Person")

 Test auf Knoten mit diesem Namen
- 3. Prädikat. Filtert die durch Achse (1.) und Knotentest (2.) spezifizierte Knotenmenge weiter. Jeder gültige XPath-Ausdruck kann Prädikat sein.

Beispiele: Test auf Kindknoten, Position bzw. Index

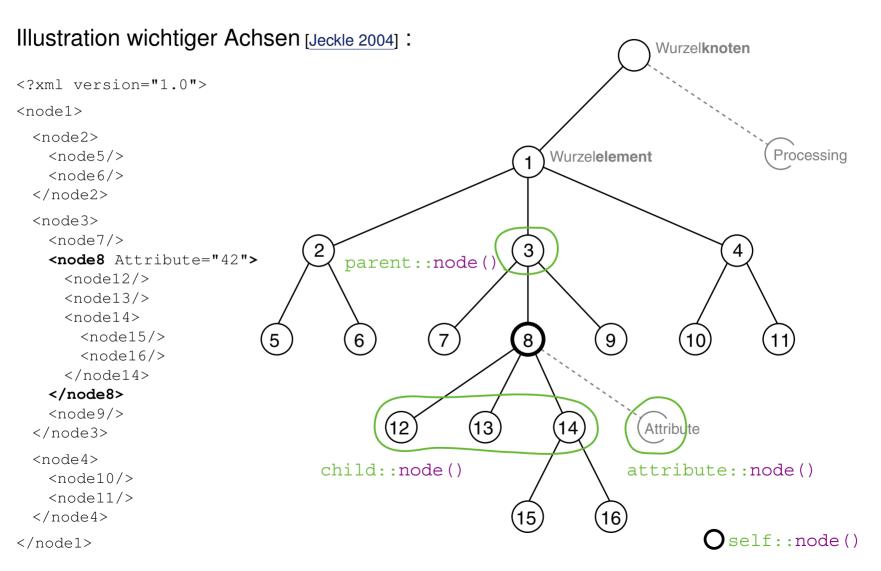
WT:III-293 Dokumentsprachen © STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)



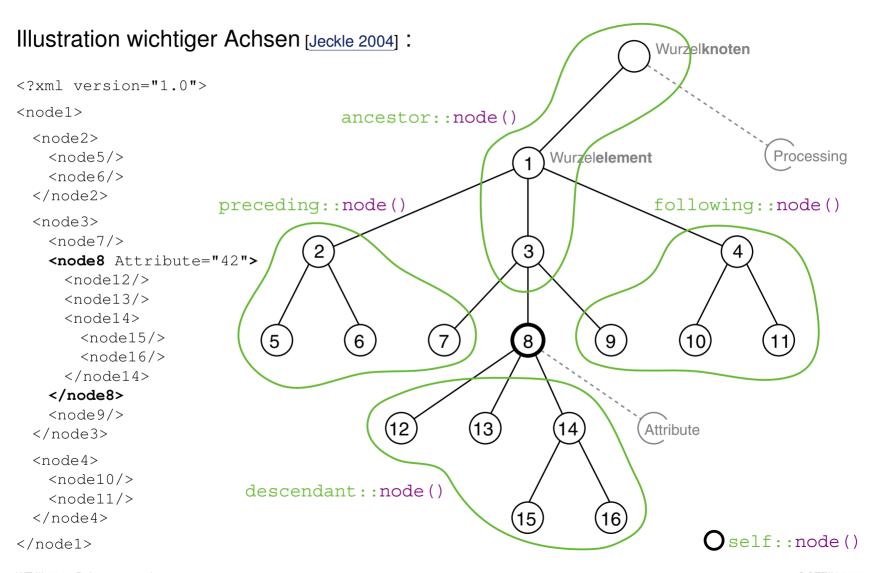
WT:III-294 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)



WT:III-295 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)



WT:III-296 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

Schreibweisen häufig verwendeter Lokalisierungsschritte:

kurz	lang	Semantik
	self::node()	Kontextknoten
	parent::node()	Elternknoten des Kontextknotens
//	<pre>/descendant-or-self::node()/</pre>	Kontextknoten einschließlich aller seiner Nachkommen
@ *	attribute::*	alle Attributknoten

Wildcards für Knotentests

```
mode()

knoten jedes Typs

Element- oder Attributknoten (achensabhängig)

Textknoten

comment()

processing-instruction()

Knoten jedes Typs

Element- oder Attributknoten (achensabhängig)

Textknoten

Kommentarknoten

Verarbeitungsanweisungsknoten
```

Spezifikation von alternativen Lokalisierungspfaden

Pfad 1 | Pfad 2 | ... | Pfad n

XPath-Lokalisierungspfade (Fortsetzung)

□ Schreibweisen häufig verwendeter Lokalisierungsschritte:

kurz	lang	Semantik
•	self::node()	Kontextknoten
	parent::node()	Elternknoten des Kontextknotens
//	<pre>/descendant-or-self::node()/</pre>	Kontextknoten einschließlich aller seiner Nachkommen
@ *	attribute::*	alle Attributknoten

Wildcards für Knotentests:

node()	Knoten jedes Typs
*	Element- oder Attributknoten (achensabhängig)
text()	Textknoten
comment()	Kommentarknoten
<pre>processing-instruction()</pre>	Verarbeitungsanweisungsknoten

Spezifikation von alternativen Lokalisierungspfaden

```
Pfad 1 | Pfad 2 | ... | Pfad n
```

WT:III-298 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

□ Schreibweisen häufig verwendeter Lokalisierungsschritte:

kurz	lang	Semantik
•	self::node()	Kontextknoten
	parent::node()	Elternknoten des Kontextknotens
//	<pre>/descendant-or-self::node()/</pre>	Kontextknoten einschließlich aller seiner Nachkommen
@ *	attribute::*	alle Attributknoten

Wildcards für Knotentests:

node()	Knoten jedes Typs
*	Element- oder Attributknoten (achensabhängig)
text()	Textknoten
comment()	Kommentarknoten
<pre>processing-instruction()</pre>	Verarbeitungsanweisungsknoten

Spezifikation von alternativen Lokalisierungspfaden:

```
Pfad 1 | Pfad 2 | ... | Pfad n
```

WT:III-299 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

(a) //person/name/descendant::*

WT:III-300 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
    <vorname>Alan
    <nachname>Turing</nachname>
    <vorname>Judea
    <nachname>Pearl
```

Beispiele für Lokalisierungspfade:

(a) //person/name/descendant::*

WT:III-301 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

- (a) //person/name/descendant::*
- (b) //geburtstag/parent::*/name

WT:III-302 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
  <name>
    <vorname>Alan
    <nachname>Turing</nachname>
  </name>
  <name>
    <vorname>Judea
    <nachname>Pearl
  </name>
```

Beispiele für Lokalisierungspfade:

- (a) //person/name/descendant::*
- (b) //geburtstag/parent::*/name

WT:III-303 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

- (a) //person/name/descendant::*
- (b) //geburtstag/parent::*/name
- (c) /personen/child::name

WT:III-304 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
```

Beispiele für Lokalisierungspfade:

- (a) //person/name/descendant::*
- (b) //geburtstag/parent::*/name
- (c) /personen/child::name

WT:III-305 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

- (a) //person/name/descendant::*
- (b) //geburtstag/parent::*/name
- (C) /personen/child::name
- (d) /personen/descendant::name

WT:III-306 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
  <name>
    <vorname>Alan
    <nachname>Turing</nachname>
  </name>
  <name>
    <vorname>Judea
    <nachname>Pearl
  </name>
```

Beispiele für Lokalisierungspfade:

- (a) //person/name/descendant::*
- (b) //geburtstag/parent::*/name
- (C) /personen/child::name
- (d) /personen/descendant::name

WT:III-307 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Beispiele für Lokalisierungspfade:

```
(a) //person/name/descendant::*
```

```
(b) //geburtstag/parent::*/name
```

- (C) /personen/child::name
- (d) /personen/descendant::name
- (e) //person[geburtstag!='unknown']

WT:III-308 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
```

Beispiele für Lokalisierungspfade:

- (a) //person/name/descendant::*
- (b) //geburtstag/parent::*/name
- (C) /personen/child::name
- (d) /personen/descendant::name
- (e) //person[geburtstag!='unknown']

WT:III-309 Dokumentsprachen ©STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

Algorithmus zur Auswertung eines Lokalisierungspfades:

```
... / Schritt_i / Schritt_i+1 / ... \downarrow M_i M_{i+1}
```

- 1. Die Auswertung der Lokalisierungsschritte geschieht von links nach rechts.
- 2. Jeder Lokalisierungsschritt spezifiziert eine Knotenmenge M.

WT:III-310 Dokumentsprachen © STEIN 2022

XPath-Lokalisierungspfade (Fortsetzung)

Algorithmus zur Auswertung eines Lokalisierungspfades:

$$...$$
 / Schritt_i / Schritt_i+1 / $...$ \downarrow \downarrow M_i M_{i+1}

- 1. Die Auswertung der Lokalisierungsschritte geschieht von links nach rechts.
- 2. Jeder Lokalisierungsschritt spezifiziert eine Knotenmenge M.
- 3. Jeder Knoten n der Knotenmenge M_i des Lokalisierungsschritts i wird als Kontextknoten hinsichtlich des Lokalisierungsschritts i+1 interpretiert und spezifiziert im Lokalisierungsschritt i+1 die Knotenmenge M_{in} .
- 4. Die Vereinigung der Mengen M_{i_n} , $n \in M_i$, bildet die Knotenmenge M_{i+1} des Lokalisierungsschritts i+1.

WT:III-311 Dokumentsprachen © STEIN 2022

Bemerkungen:

- □ Jeder der in irgendeinem Schritt spezifizierten Knoten kommt im Laufe der Auswertung in die Rolle des Kontextknotens.
- □ Vergleich verschiedener Lokalisierungspfade am Beispiel:
 - 1. Alle <beruf>-Elemente:

```
/descendant-or-self::node()/beruf (bzw.//beruf)

=
/descendant-or-self::beruf
```

2. Von jedem Elementknoten das jeweils zweite <beruf>-Kindelement:

```
/descendant-or-self::node()/beruf[2] (bzw.//beruf[2])

#
/descendant-or-self::beruf[2]
(das zweite <beruf>-Element aus dem Dokument)
```

WT:III-312 Dokumentsprachen © STEIN 2022

Lokalisierungspfade (Fortsetzung)

Knoten

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
     <vorname>Alan</vorname>
     <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912</qeburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
     <vorname>Judea
     <nachname>Pearl</nachname>
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Illustration des Algorithmus:

```
//person/name/descendant::* //person/name/descendant::*  M_1 = \{1\}   M_2 = \{1, \dots, 17\}   M_3 = \{5, 12\}   M_4 = \{6, 13\}   M_5 = \{7, 8, 14, 15\}
```

WT:III-313 Dokumentsprachen ©STEIN 2022

Lokalisierungspfade (Fortsetzung)

Knoten

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ...?>
<personen>
 <person>
   <name>
     <vorname>Alan</vorname>
     <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912</qeburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
     <vorname>Judea
     <nachname>Pearl</nachname>
   </name>
   <qeburtstag>unknown/qeburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Illustration des Algorithmus:

```
//person/name/descendant::*
//person/name/descendant::*
M_1 = \{1\}
M_2 = \{1, \dots, 17\}
M_3 = \{5, 12\}
M_4 = \{6, 13\}
M_5 = \{7, 8, 14, 15\}
```

WT:III-314 Dokumentsprachen ©STEIN 2022

Lokalisierungspfade (Fortsetzung)

Knoten

```
2 <?xml version="1.0" ?>
 3 <?xml-stylesheet type="text/xsl" ...?>
  <personen>
    <person>
      <name>
 6
       <vorname>Alan
       <nachname>Turing</nachname>
      </name>
      <geburtstag>23. Juni 1912</qeburtstag>
      <beruf>Mathematiker</peruf>
10
      <beruf>Informatiker</peruf>
11
    </person>
12
    <person>
13
      <name>
       <vorname>Judea
14
       <nachname>Pearl</nachname>
15
13
      </name>
      <qeburtstag>unknown/qeburtstag>
16
17
      <beruf>Informatiker</peruf>
12
    </person>
 4 </personen>
```

Illustration des Algorithmus:

```
\begin{tabular}{ll} $//person/name/descendant::*\\ $//person/name/descendant::*\\ $M_1=\{1\}$\\ $M_2=\{1,\ldots,17\}$\\ $M_3=\{5,12\}$\\ $M_4=\{6,13\}$\\ $M_5=\{7,8,14,15\}$\\ \end{tabular}
```

WT:III-315 Dokumentsprachen ©STEIN 2022

Lokalisierungspfade (Fortsetzung)

Knoten

```
<?xml version="1.0" ?>
  <?xml-stylesheet type="text/xsl" ...?>
  <personen>
    <person>
      <name>
        <vorname>Alan</vorname>
        <nachname>Turing</nachname>
      </name>
      <geburtstag>23. Juni 1912</qeburtstag>
      <beruf>Mathematiker</peruf>
      <beruf>Informatiker</peruf>
    </person>
 5
12
    <person>
      <name>
        <vorname>Judea
        <nachname>Pearl</nachname>
      </name>
      <qeburtstag>unknown/qeburtstag>
      <beruf>Informatiker</peruf>
12
    </person>
  </personen>
```

Illustration des Algorithmus:

```
//person/name/descendant::*
//person/name/descendant::*
M_1 = \{1\}
M_2 = \{1, \dots, 17\}
M_3 = \{5, 12\}
M_4 = \{6, 13\}
M_5 = \{7, 8, 14, 15\}
```

WT:III-316 Dokumentsprachen ©STEIN 2022

Lokalisierungspfade (Fortsetzung)

Knoten

```
<?xml version="1.0" ?>
  <?xml-stylesheet type="text/xsl" ...?>
  <personen>
    <person>
      <name>
 6
        <vorname>Alan</vorname>
        <nachname>Turing</nachname>
      </name>
      <geburtstag>23. Juni 1912</qeburtstag>
      <beruf>Mathematiker</peruf>
      <beruf>Informatiker</peruf>
    </person>
    <person>
13
      <name>
        <vorname>Judea
        <nachname>Pearl</nachname>
13
      </name>
      <qeburtstag>unknown/qeburtstag>
      <beruf>Informatiker</peruf>
    </person>
  </personen>
```

Illustration des Algorithmus:

```
//person/name/descendant::*
//person/name/descendant::*
M_1 = \{1\}
M_2 = \{1, \dots, 17\}
M_3 = \{5, 12\}
M_4 = \{6, 13\}
M_5 = \{7, 8, 14, 15\}
```

WT:III-317 Dokumentsprachen ©STEIN 2022

Lokalisierungspfade (Fortsetzung)

Knoten

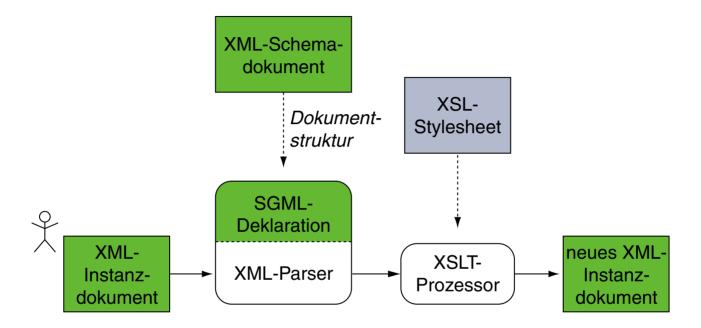
```
<?xml version="1.0" ?>
  <?xml-stylesheet type="text/xsl" ...?>
  <personen>
    <person>
      <name>
        <vorname>Alan</vorname>
        <nachname>Turing</nachname>
      </name>
      <geburtstag>23. Juni 1912/geburtstag>
      <beruf>Mathematiker</peruf>
      <beruf>Informatiker</peruf>
    </person>
    <person>
      <name>
        <vorname>Judea
14
        <nachname>Pearl</nachname>
15
      </name>
      <qeburtstag>unknown/qeburtstag>
      <beruf>Informatiker</peruf>
    </person>
  </personen>
```

Illustration des Algorithmus:

```
\label{eq:mass_mass_descendant::*} $$ \sline / person/name/descendant::* $$ \sline M_1 = \{1\} $$ \sline M_2 = \{1, \dots, 17\} $$ \sline M_3 = \{5, 12\} $$ \sline M_4 = \{6, 13\} $$ \sline M_5 = \{7, 8, 14, 15\} $$
```

WT:III-318 Dokumentsprachen ©STEIN 2022

XSL Transformation



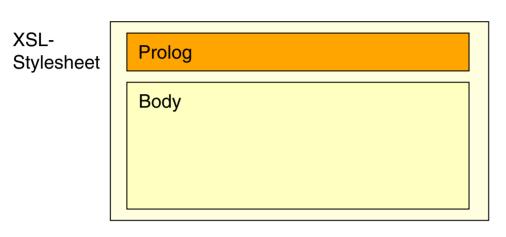
XSLT ist eine <u>Turing-vollständige</u> Programmiersprache zur Transformation wohlgeformter XML-Dokumente in andere XML-Dokumente. Ein XSLT-Programm liegt üblicherweise als XSL-Stylesheet vor.

Die Transformation umfasst die Selektion von Teilen des Eingabedokuments, deren Umordnung sowie die Generierung neuer Inhalte aus den bestehenden.

WT:III-319 Dokumentsprachen ©STEIN 2022

Aufbau XSL-Stylesheet

XSL-Stylesheets sind XML-Dokumente:

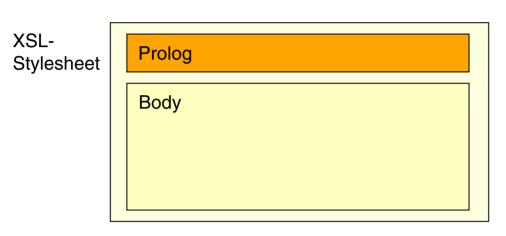


- Wurzelelement jedes XSL-Schemas ist das Element <xsl:stylesheet> oder synonym <xsl:transform>.
- □ Die Kindelemente von <xsl:stylesheet> bzw. <xsl:transform> definieren Transformationsvorschriften in Form von Template-Regeln.
- Vergleiche hierzu die XML-Dokumentstruktur und die XML-Schema-Dokumentstruktur.

WT:III-320 Dokumentsprachen © STEIN 2022

Aufbau XSL-Stylesheet

XSL-Stylesheets sind XML-Dokumente:



- Wurzelelement jedes XSL-Schemas ist das Element <xsl:stylesheet> oder synonym <xsl:transform>.
- Vergleiche hierzu die XML-Dokumentstruktur und die XML-Schema-Dokumentstruktur.

WT:III-321 Dokumentsprachen © STEIN 2022

Bemerkungen:

□ Das XSLT-Vokabular enthält die Namen für Elemente, die zur Erstellung von XSL-Stylesheets zur Verfügung stehen:

Der zugehörige Namensraum heißt http://www.w3.org/1999/XSL/Transform. Das ubliche
Präfix bei der Namensraumdeklaration ist xsl:, es kann aber beliebig gewählt werden. Wird der offizielle Namensraum gebunden, ist auch das Attribut <a href="mailto:version="1.0" anzugeben.

- □ Die Dateiendung einer XSL-Stylesheet-Datei ist .xsl.
- Aufbau einer realen Turingmaschine. [youtube]

WT:III-322 Dokumentsprachen ©STEIN 2022

XML-Beispieldokument

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="personen2html.xsl"?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <qeburtstaq>unknown/qeburtstaq>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-323 Dokumentsprachen ©STEIN 2022

Bemerkungen:

- □ Die Verknüpfung von XML-Dokument und XSL-Stylesheet kann explizit, in Form von Parametern für den XSLT-Prozessor, aber auch implizit geschehen:
 - Die Zeile <?xml-stylesheet type="text/xsl" href="..."?> im Prolog eines XML-Dokuments deklariert ein Stylesheet. Vergleiche hierzu die Stylesheet-Deklaration in HTML-Dokumenten.
- □ Beispiel: Verknüpfung von <u>personen.xml</u> (1) mit einem <u>Stylesheet</u> (2) zu einem <u>HTML-Dokument</u> (3). Der Quelltext (→ Seitenquelltext im Browser) der Dateien (1) und (3) unterscheidet sich nur um die Zeile, mit der das Stylesheet eingebunden wird.
- □ Aufruf des XSLT-Prozessors Xalan-J über die Kommandozeile:

```
java org.apache.xalan.xslt.Process -in personen.xml -xsl tiny.xsl
```

Hierfür muss die <u>Xalan-Bibliothek</u> heruntergeladen und der Ort der Bibliothek im Classpath spezifiziert sein. Alternativ der Aufruf mit expliziter Angabe der Xalan-Bibliothek:

```
java -cp <path>/xalan.jar ...
```

□ Aufruf des XSLT-Prozessor via Microsoft Visual Studio: Dokumentation

WT:III-324 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets

Das einfachste (= leere) Stylesheet:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
</xsl:stylesheet>
```

WT:III-325 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets

Das einfachste (= leere) Stylesheet:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
</xsl:stylesheet>
```

Angewandt auf das Beispieldokument:

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

Alan
Turing

23. Juni 1912
Mathematiker
Informatiker

Judea
Pearl

unknown
Informatiker
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
    <vorname>Judea</vorname>
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker
 </person>
</personen>
```

WT:III-326 Dokumentsprachen ©STEIN 2022

Bemerkungen:

- □ Das leere Stylesheet in dem Bespiel enthält keine matchende Template-Regel. Die Ausgabe entsteht, weil in einer solchen Situation vom XSLT-Prozessor das Built-in-Template zur Ausgabe von Text- und Attributknoten angewandt wird.
- □ Diejenigen Konstrukte eines XML-Dokuments, die nicht zu einem der sieben Knotentypen des XPath-Modells gehören, werden unverändert übernommen. Hierzu zählt u.a. die <?xml ...?>-Deklaration.

WT:III-327 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Wichtigstes Stylesheet-Element ist die Template-Regel (*Template*):

- ightharpoonup Der Lokalisierungspfad des match-Attributs spezifiziert ausgehend von dem Kontextknoten eine Knotenmenge M.
- \Box Wird während der Verarbeitung eines XML-Dokuments ein Knoten in M erreicht, dann matched die Template-Regel diesen Knoten.

WT:III-328 Dokumentsprachen © STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Wichtigstes Stylesheet-Element ist die Template-Regel (*Template*):

- ightharpoonup Der Lokalisierungspfad des match-Attributs spezifiziert ausgehend von dem Kontextknoten eine Knotenmenge M.
- \Box Wird während der Verarbeitung eines XML-Dokuments ein Knoten in M erreicht, dann matched die Template-Regel diesen Knoten.
- Matched eine Template-Regel einen Knoten n, behandelt das Ersetzungsmuster den gesamten Teilbaum des XML-Dokuments, der Knoten n als Wurzel hat. Dieser Teilbaum gilt als abgearbeitet.

WT:III-329 Dokumentsprachen © STEIN 2022

Bemerkungen:

- Der Wert des match-Attributes im <xsl:template>-Element ist ein Lokalisierungspfad in eingeschränkter XPath-Syntax.
- Um eine bestimmte Knotenmenge M zu spezifizieren für deren Elemente eine Template-Regel matched, sind alternative Pfadangaben möglich. Beispielsweise spezifizieren die Ausdrücke match="// Elementname" und match="Elementname" dieselbe Knotenmenge.

D.h., ein relativer Lokalisierungspfad des <xsl:template>-Elements kann wie der entsprechende absolute durch "//" eingeleitete Lokalisierungspfad aufgefasst werden – und umgekehrt.

WT:III-330 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet mit konstanter Textausgabe:

Angewandt auf das Beispieldokument:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl</nachname>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker
 </person>
</personen>
```

WT:III-331 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet mit konstanter Textausgabe:

Angewandt auf das Beispieldokument:

```
Person found!
```

Person found!

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl</nachname>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker
 </person>
</personen>
```

WT:III-332 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet zum Kopieren der Elemente:

Angewandt auf das Beispieldokument:

```
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl</nachname>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker
 </person>
</personen>
```

WT:III-333 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet zum Kopieren der Elemente:

Angewandt auf das Beispieldokument:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
     <vorname>Alan</vorname>
     <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
     <vorname>Judea
     <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-334 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet zum Kopieren der Elementinhalte:

Angewandt auf das Beispieldokument:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl</nachname>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker
 </person>
</personen>
```

WT:III-335 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet zum Kopieren der Elementinhalte:

Angewandt auf das Beispieldokument:

```
Alan
Turing

23. Juni 1912
Mathematiker
Informatiker
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-336 Dokumentsprachen © STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet zur Elementselektion mittels leerer Template-Regeln:

Angewandt auf das Beispieldokument:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-337 Dokumentsprachen ©STEIN 2022

Elemente eines XSL-Stylesheets (Fortsetzung)

Stylesheet zur Elementselektion mittels leerer Template-Regeln:

Angewandt auf das Beispieldokument:

```
Turing, Alan
Pearl, Judea
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Vergleiche die Elementselektion durch explizite Verarbeitungssteuerung.

WT:III-338 Dokumentsprachen ©STEIN 2022

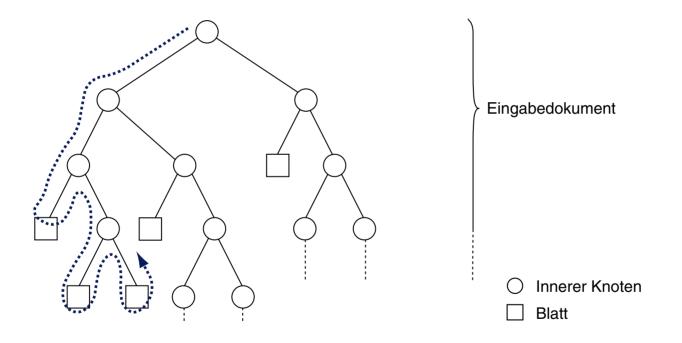
Bemerkungen: (Wiederholung)

- □ Matched eine Template-Regel einen Knoten im XML-Dokument, so gilt der Knoten einschließlich des zugehörigen Teilbaums als abgearbeitet.
- ☐ Mit leeren Template-Regeln kann man Knoten und Teilbäume filtern, die nicht in der Ausgabe erscheinen sollen.
- □ Matched keine Template-Regel des Stylesheets einen Knoten im XML-Dokument, wird vom XSLT-Prozessor das Built-in-Template zur Ausgabe von Text- und Attributknoten angewandt.

WT:III-339 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Verarbeitungsstrategie

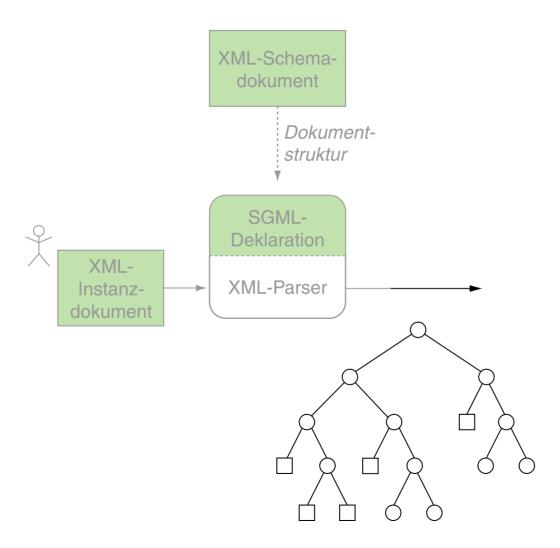
Standardmäßig durchläuft der XSLT-Prozessor den aus dem Eingabedokument erzeugten Baum ausgehend vom Wurzelknoten in Pre-Order-Reihenfolge.



Während des Traversierungsvorgangs wird für jeden besuchten Knoten das speziellste, matchende Template gesucht und angewandt. So transformiert der XSLT-Prozessor einen XML-Quellbaum in einen XML-Zielbaum.

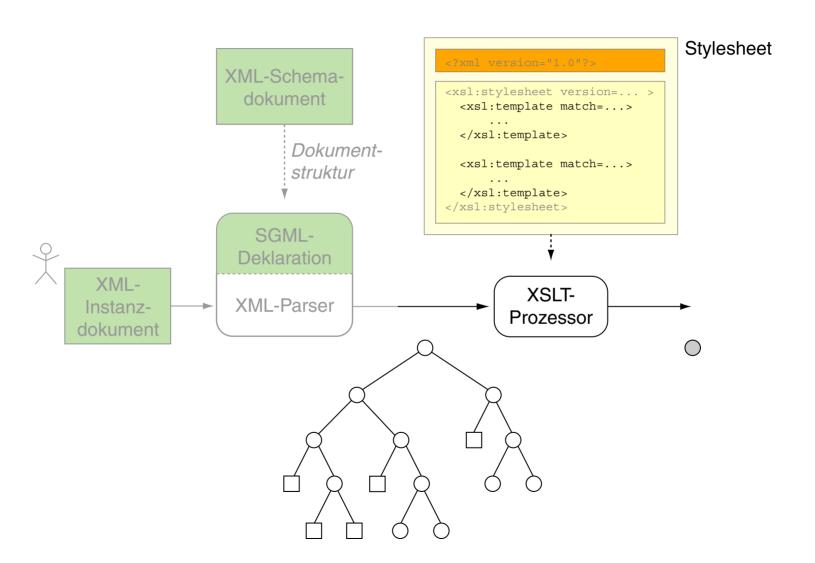
WT:III-340 Dokumentsprachen © STEIN 2022

XSLT-Prozessor: Verarbeitungsstrategie (Fortsetzung)



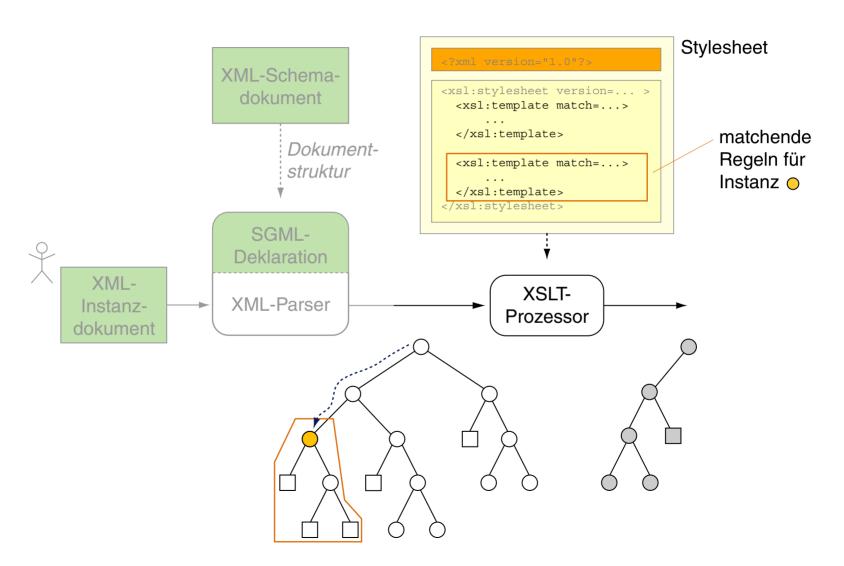
WT:III-341 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Verarbeitungsstrategie (Fortsetzung)



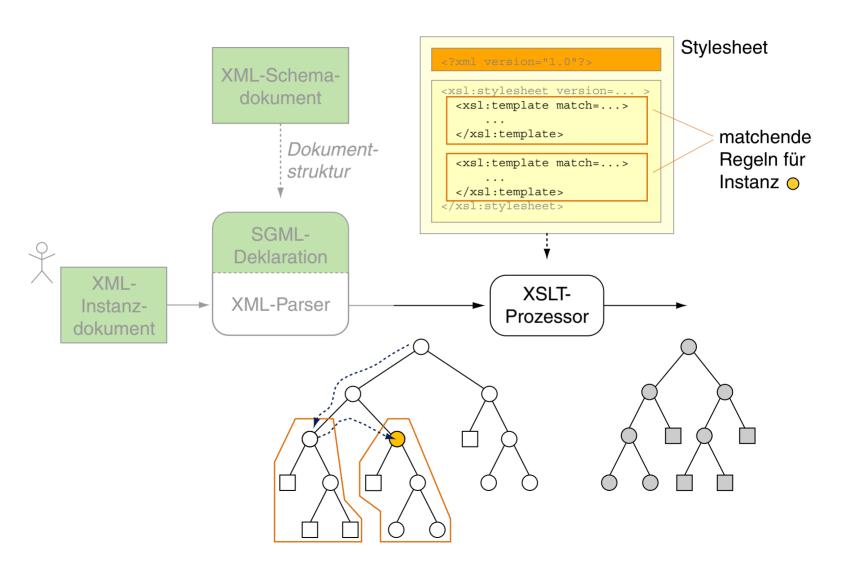
WT:III-342 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Verarbeitungsstrategie (Fortsetzung)



WT:III-343 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Verarbeitungsstrategie (Fortsetzung) [WT:III CSS-Verarbeitung]



WT:III-344 Dokumentsprachen ©STEIN 2022

Bemerkungen:

- Aus Verarbeitungssicht spielt somit die Reihenfolge der Template-Regeln in einem XSL-Stylesheet keine Rolle: die Verarbeitung wird ausschließlich durch die *Reihenfolge der Elemente im Eingabedokument* bestimmt.
- □ Ein Anwendungskonflikt zur Verarbeitung eines Knoten liegt vor, wenn Lokalisierungspfade von mehreren Template-Regeln diesen Knoten in ihrer spezifizierten Knotenmengen enthalten. In diesem Fall kommt das spezifischste Template zur Anwendung also das Template, das die kleinste Knotenmenge spezifiziert.

WT:III-345 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Verarbeitungsstrategie (Fortsetzung)

Stylesheet zur Elementselektion mit expliziter Verarbeitungssteuerung:

Angewandt auf das Beispieldokument:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-346 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Verarbeitungsstrategie (Fortsetzung)

Stylesheet zur Elementselektion mit expliziter Verarbeitungssteuerung:

Angewandt auf das Beispieldokument:

```
Turing, Alan
Pearl, Judea
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan</vorname>
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl</nachname>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Vergleiche die Elementselektion mittels leerer Template-Regeln.

WT:III-347 Dokumentsprachen ©STEIN 2022

Bemerkungen:

- Das <xsl:apply-templates>-Element startet für die mit dem select-Attribut spezifizierte Knotenmenge erneut einen Pre-Order-Durchlauf zur Anwendung der Template-Regeln des Stylesheets.
- Der Wert des select-Attributes im <xsl:apply-templates>-Element ist ein Lokalisierungspfad in eingeschränkter XPath-Syntax. Weil sich so beliebige Knoten im Dokument spezifizieren lassen, ermöglicht das <xsl:apply-templates>-Element die mehrmalige Verarbeitung von Knoten, also auch die Erzeugung von Endlosschleifen.
- □ Falls keine andere Achse angegeben ist, setzt der Lokalisierungspfad des <xsl:apply-templates>-Elements den Pfad des matchenden Knoten fort. Das heißt, die Ausdrücke select="./Elementname" und select="Elementname" spezifizieren dieselbe Knotenmenge.
- □ Enthält das <xsl:apply-templates>-Element kein select-Attribut, so gelten per Default die Kindknoten (child::-Achse) des matchenden Knoten als spezifiziert.

WT:III-348 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Beispiele zur Verarbeitungsstrategie

Stylesheet mit zweifacher Verarbeitung der <name>-Kindelemente:

Angewandt auf das Beispieldokument:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-349 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Beispiele zur Verarbeitungsstrategie

Stylesheet mit zweifacher Verarbeitung der <name>-Kindelemente:

Angewandt auf das Beispieldokument:

```
Turing, AlanTuring, Alan Pearl, JudeaPearl, Judea
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
  <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-350 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Beispiele zur Verarbeitungsstrategie (Fortsetzung)

Stylesheet zur wiederholten Verarbeitung aller <name>-Elemente:

Angewandt auf das Beispieldokument:

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-351 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Beispiele zur Verarbeitungsstrategie (Fortsetzung)

Stylesheet zur wiederholten Verarbeitung aller <name>-Elemente:

Angewandt auf das Beispieldokument:

```
Turing, AlanPearl, Judea
Turing, AlanPearl, Judea
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-352 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Beispiele zur Verarbeitungsstrategie (Fortsetzung)

Stylesheet, dessen matchende Template-Regel die leere Knotenmenge liefert:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/...">
 <xsl:template match="person">
   <xsl:apply-templates select="nachname"/>
 </xsl:template>
 <xsl:template match="name">
   <xsl:value-of select="nachname"/>
   <xsl:text>, </xsl:text>
   <xsl:value-of select="vorname"/>
 </xsl:template>
</xsl:stylesheet>
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-353 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Beispiele zur Verarbeitungsstrategie (Fortsetzung)

Stylesheet, dessen Verarbeitung in eine Endlosschleife führt:

Angewandt auf das Beispieldokument:

```
(Location of error unknown) XSLT Error (java.lang.StackOverflowError): null
```

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
    <vorname>Alan
    <nachname>Turing</nachname>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker
   <beruf>Informatiker</peruf>
 </person>
 <person>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-354 Dokumentsprachen ©STEIN 2022

XSLT-Prozessor: Built-in-Templates [xpath notation]

 Built-in-Template, das die rekursive Verarbeitung garantiert, falls kein matchendes Template im Stylesheet existiert:

```
<xsl:template match="*|/">
  <xsl:apply-templates/>
</xsl:template>
```

2. Built-in-Template zur Ausgabe von Text- und Attributknoten:

```
<xsl:template match="text()|@*">
    <xsl:value-of select="."/>
</xsl:template>
```

3. Built-in-Template, das die Kommentare matched und ignoriert:

```
<xsl:template match="processing-instruction()|comment()"/>
```

Vergleiche die Elementselektion mittels leerer Template-Regeln.

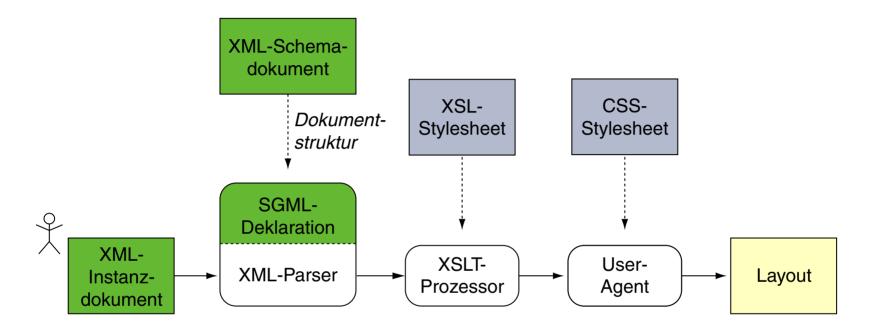
WT:III-355 Dokumentsprachen ©STEIN 2022

Weitere XSLT-Konzepte

- Template-Modi zur Charakterisierung von Verarbeitungsphasen
- benannte Templates zur Realisierung direkter Aufrufe
- Nummerierung und Sortierung von Ausgabeelementen
- bedingte Verarbeitung und Schleifen
- Import anderer Stylesheets

WT:III-356 Dokumentsprachen ©STEIN 2022

XML-Dokumentenverarbeitung: Erzeugung von HTML-Dokumenten



Vergleiche hierzu den Standardprozess der XSL Transformation.

WT:III-357 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten (Fortsetzung)

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="personen2html.xsl"?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <qeburtstaq>unknown/qeburtstaq>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-358 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten (Fortsetzung) [Eingangsbeispiel]

WT:III-359 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten (Fortsetzung) [Eingangsbeispiel]

WT:III-360 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten (Fortsetzung) [Eingangsbeispiel]

WT:III-361 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten (Fortsetzung) [Eingangsbeispiel]

```
<xsl:template match="personen">
 <html>
   <head>
     <title>
      <xsl:text>Personen</xsl:text>
     </title>
   </head>
   <body>
     <xsl:apply-templates/>
   </body>
 </html>
</xsl:template>
<xsl:template match="name">
 <div>
   <xsl:text>Name: </xsl:text>
   <span style="font-weight:bold">
     <xsl:value-of select="self::*"/>
   </span>
 </div>
</xsl:template>
```

WT:III-362 Dokumentsprachen

Erzeugung von HTML-Dokumenten (Fortsetzung)

[Eingangsbeispiel]

```
<xsl:template match="personen">
 <html>
   <head>
     <title>
       <xsl:text>Personen</xsl:text>
     </title>
   </head>
   <body>
     <xsl:apply-templates/>
   </body>
 </html>
</xsl:template>
<xsl:template match="name">
 >
 < div>
   <xsl:text>Name: </xsl:text>
   <span style="font-weight:bold">
     <xsl:value-of select="self::*"/>
   </span>
 </div>
</xsl:template>
```

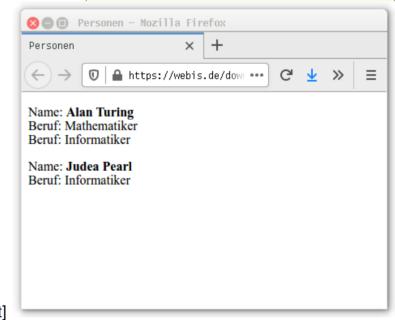
```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" ... ?>
<personen>
 <person>
   <name>
    <vorname>Alan
    <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912</geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-363 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten (Fortsetzung)

```
<xsl:template match="personen">
 <html>
   <head>
     <title>
       <xsl:text>Personen</xsl:text>
     </title>
   </head>
   <body>
     <xsl:apply-templates/>
   </body>
 </html>
</xsl:template>
<xsl:template match="name">
 >
 < div>
   <xsl:text>Name: </xsl:text>
   <span style="font-weight:bold">
     <xsl:value-of select="self::*"/>
   </span>
 </div>
</xsl:template>
                            [ohne / mit Stylesheet]
```

[Eingangsbeispiel]

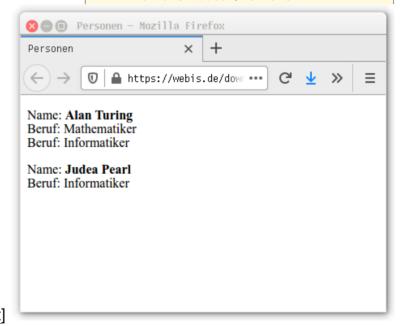


WT:III-364 Dokumentsprachen © STEIN 2022

Erzeugung von HTML-Dokumenten (Fortsetzung)

```
<xsl:template match="personen">
 <html>
   <head>
     <title>
       <xsl:text>Personen</xsl:text>
     </title>
   </head>
   <body>
     <xsl:apply-templates/>
   </body>
 </html>
</xsl:template>
<xsl:template match="name">
 >
 < div>
   <xsl:text>Name: </xsl:text>
   <span style="font-weight:bold">
     <xsl:value-of select="self::*"/>
   </span>
 </div>
</xsl:template>
                            [ohne / mit Stylesheet]
```

[Eingangsbeispiel]



Erzeugung von HTML-Dokumenten (Fortsetzung)

```
<xsl:template match="personen">
 <html>
   <head>
     <title>
      <xsl:text>Personen</xsl:text>
     </title>
     <link rel="stylesheet" type="text/css" href="personen.css"/>
   </head>
   <body>
     <xsl:apply-templates/>
   </body>
 </html>
</xsl:template>
<xsl:template match="name">
 <div>
   <xsl:text>Name: </xsl:text>
     <xsl:value-of select="self::*"/>
 </div>
</xsl:template>
```

WT:III-366 Dokumentsprachen ©STEIN 2022

Bemerkungen:

Eine Anwendung nach diesem Schema sind die FAQs des W3C: Aus der XML-Source faq.xml gemäß der DTD faq.dtd wird mittels des Stylesheets faqxsl.xsl das HTML-Dokument faq.html erzeugt.

Weil in <u>faq.xml</u> das Stylesheet <u>faq.css</u> verlinkt ist, zeigt der Browser nicht den XML-Dokumentenbaum an:

WT:III-367 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten: Datenaufbereitung

CD-Datenbank als XML-Dokument [w3schools]:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>
<catalog>
 \langle cd \rangle
   <title>Empire Burlesque</title>
   <artist>Bob Dylan</artist>
   <company>Columbia
   <price>10.90</price>
   <year>1985
 </cd>
 \langle cd \rangle
   <title>Unchain my heart</title>
   <artist>Joe Cocker</artist>
   <company>EMI</company>
   <price>8.20</price>
   <year>1987
 </cd>
</catalog>
```

WT:III-368 Dokumentsprachen © STEIN 2022

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/...">
<xsl:template match="/">
 <html>
 <body>
  <h2>Mv CD Collection</h2>
  Title
     Artist
    </t.r>
    <xsl:for-each select="catalog/cd">
    \langle tr \rangle
     <xsl:value-of select="title"/>
     <xsl:value-of select="artist"/>
    </xsl:for-each>
  </body>
 </html>
</xsl:template>
</xsl:stylesheet>
```

WT:III-369 Dokumentsprachen

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/...">
<xsl:template match="/">
 <html>
 <body>
   <h2>My CD Collection</h2>
   Title
      Artist
    </t.r>
    <xsl:for-each select="catalog/c</pre>
    \langle tr \rangle
      <xsl:value-of select="titl"
      <xsl:value-of select="arti
    </xsl:for-each>
   </body>
 </html>
</xsl:template>
                 [w3schools xml, xsl, editor]
</xsl:stylesheet>
```

My CD Collection	
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli
When a man loves a woman	Percy Sledge
Black angel	Savage Rose
1999 Grammy Nominees	Many
For the good times	Konny Rogers

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

Filtern mit XPath:

```
<xsl:template match="/">
 <ht.ml>
 <body>
  <h2>My CD Collection</h2>
  Title
    Artist
   <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
   <xsl:value-of select="title"/>
    <xsl:value-of select="artist"/>
   </xsl:for-each>
  </body>
 </html>
</xsl:template>
```

WT:III-371 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

Filtern mit XPath:

```
<xsl:template match="/">
 <ht.ml>
 <body>
  <h2>My CD Collection</h2>
  Title
     Artist
    <xsl:for-each select="catalog/cd[artist='Bob Dylan']">
    x - D Mozilla Firefox
     <xsl:value-of select="title"
     <xsl:value-of select="arti
                                My CD Collection
    </t.r>
    </xsl:for-each>
                                Title
                                            Artist
  Empire Burlesque Bob Dylan
 </body>
 </html>
</xsl:template>
                        [w3schools]
```

WT:III-372 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

Sortierte Ausgabe:

```
<xsl:template match="/">
 <ht.ml>
 <body>
  <h2>My CD Collection</h2>
  Title
     Artist
    <xsl:for-each select="catalog/cd">
    <xsl:sort select="artist"/>
    \langle tr \rangle
     <xsl:value-of select="title"/>
     <xsl:value-of select="artist"/>
    </t.r>
    </xsl:for-each>
  </body>
 </html>
</xsl:template>
```

WT:III-373 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

Sortierte Ausgabe:

```
<xsl:template match="/">
 <ht.ml>
 <body>
  <h2>My CD Collection</h2>
  Title
     Artist
    <xsl:for-each select="catalog/c"</pre>
    <xsl:sort select="artist"/>
    <xsl:value-of select="titl
     <xsl:value-of select="art:
    </t.r>
    </xsl:for-each>
  </body>
 </html>
</xsl:template>
                       [w3schools]
```



WT:III-374 Dokumentsprachen ©STEIN 2022

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

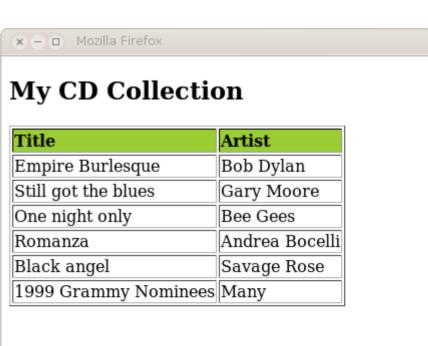
Bedingte Ausgabe:

```
<xsl:template match="/">
 <ht.ml>
 <body>
  <h2>My CD Collection</h2>
  Title
     Artist
    <xsl:for-each select="catalog/cd">
    <xsl:if test="price &qt; 10">
     \langle tr \rangle
      <xsl:value-of select="title"/>
      <xsl:value-of select="artist"/>
     </xsl:if>
    </xsl:for-each>
  </body>
 </html>
</xsl:template>
```

Erzeugung von HTML-Dokumenten: Datenaufbereitung (Fortsetzung)

Bedingte Ausgabe:

```
<xsl:template match="/">
 <ht.ml>
 <body>
  <h2>My CD Collection</h2>
  Title
     Artist
    <xsl:for-each select="catalog/colored"</pre>
    <xsl:if test="price &qt; 10">
     >
       <xsl:value-of select="til
       <xsl:value-of select="ar
     </t.r>
    </xsl:if>
    </xsl:for-each>
  </body>
 </html>
                        [w3schools]
</xsl:template>
```



XML-Dokumentenverarbeitung: Elementinhalte anpassen [wt:III DOM-API]

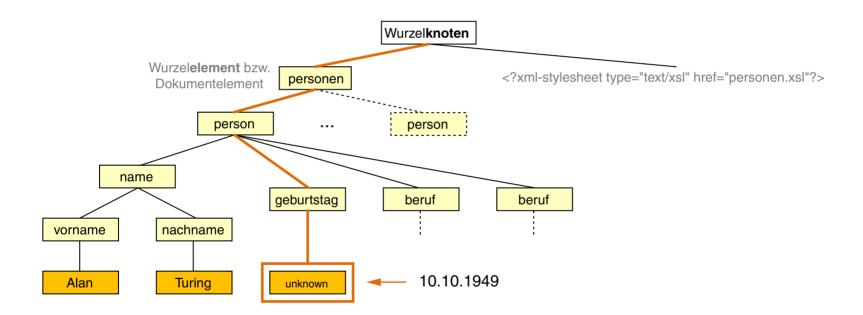
```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="personen2html.xsl"?>
<personen>
 <person>
   <name>
    <vorname>Alan
     <nachname>Turing</nachname>
   </name>
   <geburtstag>23. Juni 1912/geburtstag>
   <beruf>Mathematiker</peruf>
   <beruf>Informatiker</peruf>
 </person>
 <person>
   <name>
    <vorname>Judea
    <nachname>Pearl</nachname>
   </name>
   <qeburtstaq>unknown/qeburtstaq>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

WT:III-377 Dokumentsprachen ©STEIN 2022

XML-Dokumentenverarbeitung: Elementinhalte anpassen (Fortsetzung)

Aufgabe:

- Die Person "Judea Pearl" finden.
- 2. Seinen Geburtstag auf einen bestimmten Wert setzen.



WT:III-378 Dokumentsprachen ©STEIN 2022

XML-Dokumentenverarbeitung: Elementinhalte anpassen (Fortsetzung)

Stylesheet:

</xsl:stylesheet>

WT:III-379 Dokumentsprachen ©STEIN 2022

XML-Dokumentenverarbeitung: Elementinhalte anpassen (Fortsetzung)

Stylesheet:

WT:III-380 Dokumentsprachen ©STEIN 2022

XML-Dokumentenverarbeitung: Elementinhalte anpassen (Fortsetzung)

Stylesheet:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/...">
  <xsl:template match="@*|node()"> [xpath notation]
    <xsl:copy><xsl:apply-templates select="@*|node()|"/></xsl:copy> [W3C]
  </xsl:template>
  <xsl:template match= "person[name/nachname='Pearl' and</pre>
       name/vorname='Judea']/geburtstag/text()">
    < xs1:text > 10.10.1949 < / xs1:text >
                                                   <?xml version="1.0" ?>
                                                   <?xml-stylesheet type="text/xsl" ... ?>
  </xsl:template>
                                                   <personen>
                                                    <person>
</xsl:stylesheet>
                                                       <vorname>Alan</vorname>
                                                       <nachname>Turing</nachname>
Angewandt auf das Beispieldokument:
                                                     <geburtstag>23. Juni 1912/geburtstag>
```

```
<name>
 <vorname>Judea
 <nachname>Pearl</nachname>
</name>
<geburtstag>10.10.1949/geburtstag>
```

```
<beruf>Mathematiker
   <beruf>Informatiker
 </person>
 <person>
   <name>
    <vorname>Judea</vorname>
    <nachname>Pearl</nachname>
   </name>
   <geburtstag>unknown</geburtstag>
   <beruf>Informatiker</peruf>
 </person>
</personen>
```

Quellen zum Nachlernen und Nachschlagen im Web: Referenz

- □ W3C. XSL Transformations (XSLT) 2.0. www.w3.org/TR/xslt20
- □ W3C *XML* Path Language (XPath) 3.1. www.w3.org/TR/xpath-31
- □ W3C XML Query Language (XQuery) 3.1 www.w3.org/TR/xquery-31

WT:III-382 Dokumentsprachen ©STEIN 2022

Quellen zum Nachlernen und Nachschlagen im Web: Usage

- □ Apache. *Xalan Project.* xalan.apache.org
- MDN. XSLTProcessor. developer.mozilla.org/en-US/docs/Web/API/XSLTProcessor
- Saxonica.com. XSLT and XQuery Processing. www.saxonica.com
- □ W3 Schools. XSLT.www.w3schools.com/xml/xsl_intro.asp

WT:III-383 Dokumentsprachen ©STEIN 2022