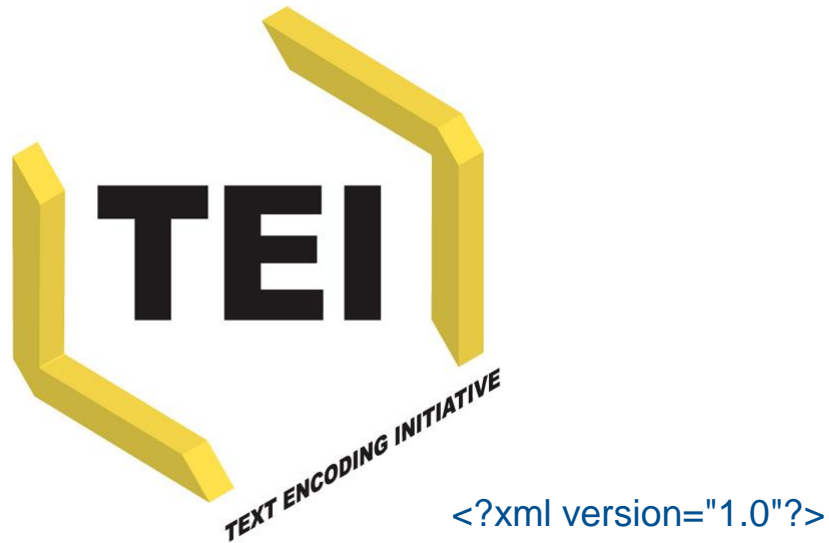


Einführung in XPath



Agnes Brauer
a.brauer@ub.uni-frankfurt.de

Vorbereitung

- Melden Sie sich für den moodle-Kurs Praxislabor Digitale Geisteswissenschaften an und schreiben Sie sich ein:

Praxislabor Digitale Geisteswissenschaften: Einführungsworkshops zu Methoden und Werkzeugen der Digital Humanities

Praxislabor Digitale Geisteswissenschaften: Einführungsworkshops zu Methoden und Werkzeugen der Digital Humanities

Zum ersten Kennenlernen von Methoden und Werkzeugen der Digital Humanities bietet die Universitätsbibliothek JCS (im Bibliothekszentrum Geisteswissenschaften) Studierenden und Mitarbeiterinnen der Goethe-Uni im kommenden Wintersemester Workshops an. In niederschweligen Einführungen werden anhand von überschaubaren, konkreten Beispielen aus der Praxis Methoden, Tools oder Themen der digitalen Geisteswissenschaften vorgestellt und geübt und so ein erster Einblick in die Möglichkeiten gegeben, wie klassische Methoden der Geisteswissenschaften mithilfe digitaler Verfahren der Textanalyse sowie der Text- und Datenaufbereitung sinnvoll ergänzt werden können.

Die Workshopreihe besteht jeweils aus inhaltlich zusammenhängenden Zweierblöcken, in denen auf eine Präsentation eine Sitzung zur Vertiefung und Übung folgt.

Die Workshops richten sich an interessierte Einsteiger; besondere Kenntnisse werden nicht vorausgesetzt. Nähere Informationen sowie die Möglichkeit zur Anmeldung finden Sie unter: <http://www.ub.uni-frankfurt.de/digitalhumanities/workshops.html>.

Praxislabor Digitale Geisteswissenschaften: Einführungsworkshops zu Methoden und Werkzeugen der Digital Humanities

Startseite / Kurse / Verschiedenes / Praxislabor Digitale Geisteswissenschaften

Allgemeines

In dieses kollaborative Dokument können Themenvorschläge für die Hands-on-Sessions eingetragen werden. Informationen zur Anmeldung und Kurszeiten unter: <http://www.ub.uni-frankfurt.de/digitalhumanities>

Einführung in TEI / XML

Dozentin: Agnes Brauer

Der Workshop führt in die Grundlagen der Textauszeichnung mit TEI ein, einer XML-basierten und sich mittlerweile als De-facto-Standard etablierten Auszeichnungssprache speziell für die Zwecke der Geisteswissenschaften. Nach einer knappen allgemeinen Einführung werden die Teilnehmer anhand einer kleinen Übung die Praxis der Textauszeichnung mit TEI kennenlernen und sich einen ersten Überblick über die Bedeutung und die verschiedenen Module dieser Sprache verschaffen.

Link: <http://www.tei-c.org/>

Hands-on Übung zur TEI/XML-Einführung

Dozentin: Agnes Brauer

Einführungsworkshops zu Methoden und Werkzeugen der Digital Humanities

Praxislabor Digitale Geisteswissenschaften: Einführungsworkshops zu Methoden und Werkzeugen der Digital Humanities

Zum ersten Kennenlernen von Methoden und Werkzeugen der Digital Humanities bietet die Universitätsbibliothek JCS (im Bibliothekszentrum Geisteswissenschaften) Studierenden und Mitarbeiterinnen der Goethe-Uni im kommenden Wintersemester Workshops an. In niederschweligen Einführungen werden anhand von überschaubaren, konkreten Beispielen aus der Praxis Methoden, Tools oder Themen der digitalen Geisteswissenschaften vorgestellt und geübt und so ein erster Einblick in die Möglichkeiten gegeben, wie klassische Methoden der Geisteswissenschaften mithilfe digitaler Verfahren der Textanalyse sowie der Text- und Datenaufbereitung sinnvoll ergänzt werden können.

Die Workshopreihe besteht jeweils aus inhaltlich zusammenhängenden Zweierblöcken, in denen auf eine Präsentation eine Sitzung zur Vertiefung und Übung folgt.

Die Workshops richten sich an interessierte Einsteiger; besondere Kenntnisse werden nicht vorausgesetzt. Nähere Informationen sowie die Möglichkeit zur Anmeldung finden Sie unter: <http://www.ub.uni-frankfurt.de/digitalhumanities/workshops.html>

Trainerin: Agnes Brauer
Trainerin: Jakob Frohmann

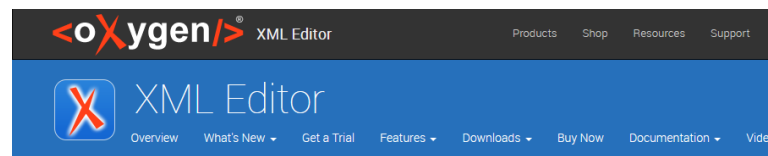
Selbsteinschreibung (Teilnehmer/in)

Kein Einschreibeschlüssel notwendig

EINSCHREIBEN

Vorbereitung

- Installieren Sie bitte [diesen XML-Editor](#):




Download Oxygen XML Editor

Download Oxygen XML Editor and test it for free for 30 days (requires a [trial license key](#)). This allows you to evaluate the software for free before making a purchase decision.

There's a single installation kit for all editions. Different license types activate different features. For a comparison between the Enterprise and Professional/Academic editions, please see the [Feature Matrix](#).

If you're looking for an older version of Oxygen XML Editor, [click here](#).

	Windows	Mac OS X	Linux	Other	Eclipse
	Windows 64-bit (Includes Java SE 8u172)				
	Version: 20.1 (See release notes) Original release date: June 15, 2018 Build id: 2018101517 Build date: October 15, 2018 Size: 244MB MD5				
	Windows 32-bit (Includes Java SE 8u172)				
	Version: 20.1 (See release notes)				

Beispiel einer XML-Datei

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xmlns="http://www.tei-c.org/ns/1.0">
  <teiHeader>
    <fileDesc>
      <titleStmt>
        <title>TEI-Minimal-Beispiel</title>
      </titleStmt>
      <publicationStmt>
        <p>Frei verfügbar</p>
      </publicationStmt>
      <sourceDesc>
        <p>Dieser Text ist digital born.</p>
      </sourceDesc>
    </fileDesc>
  </teiHeader>
  <text> <!-- Ein XML Kommentar -->
    <body>
      <p>Ein Beispieltext von <name>Agnes Brauer</name><lb/>
        für die Übung<hi rend="italic">Textauszeichnung mit TEI</hi>.</p>
    </body>
  </text>
</TEI>
```

XPath

- *XPath* dient der **Suche und Navigation** innerhalb von XML-Dokumenten
- die dritte Version von 2017 ist die momentan aktuelle
- *XPath*-Ausdrücke lokalisieren Teile eines XML-Dokuments und lesen ihre Eigenschaften aus
- darüber hinaus definiert *XPath* eine Reihe von Funktionen, die zusätzliche Operationen auf Ergebnissen von Anfragen erlauben
- *XPath* kann nicht in das XML-Dokument eingreifen
- die Sprache wird zur Ermittlung von Teilbäumen und Eigenschaften von XML-Dokumenten für viele XML-Technologien verwendet
- Beispiele für solche Technologien sind *XQuery* oder **XSLT**

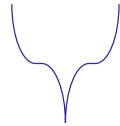
XPath

- Ein XPath besteht aus einem oder mehreren **Pfadabschnitten** (location steps)
- die Pfadabschnitte bestehen aus einem Schrägstrich (/) und einem **Knotentest** (node test)
- dem Knotentest kann eine **Achse** (axis) vorangestellt werden
- die Ergebnismenge eines Pfadabschnitts kann durch **Bedingungen** (predicates) eingeschränkt werden
- der letzte angegebene Knotentest im XPath bestimmt den **Typ** des Ergebnisses

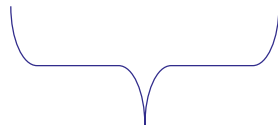
XPath

Beispiel für einen XPath:

/TEI / text / body / div / p



Knotentest
(node test)



/ Pfadabschnitt
(location step)

XPath

Knotentypen:

- **Element:** geprüft über `element()`, `element(Knotennamen)`, **Knotennamen** oder `*` (als Abkürzung für ein beliebiges Element)
- **Attribut:** geprüft über `attribute()`, `attribute(Knotennamen)`, **@Knotennamen** oder **@***
- **Text:** geprüft durch **text()**
- **Kommentar:** geprüft durch **comment()**

XPath

Übung

- Finden Sie alle Paragraphen in Divisions

XPath

Übung

- Finden Sie alle Paragraphen in Divisions

`/TEI / text / body / div / p`

XPath

Übung

- Finden Sie alle Paragraphen in Divisions
/TEI / text / body / div / p
- Finden Sie die Personennamen innerhalb von Paragraphen

XPath

Übung

- Finden Sie alle Paragraphen in Divisions
`/TEI / text / body / div / p`
- Finden Sie die Personennamen innerhalb von Paragraphen
`/TEI / text / body / div / p / persName`

XPath

Übung

- Finden Sie alle Paragraphen in Divisions
`/TEI / text / body / div / p`
- Finden Sie die Personennamen innerhalb von Paragraphen
`/TEI / text / body / div / p / persName`
- Welches Ergebnis hat der Ausdruck?
`/ TEI /teiHeader / fileDesc / titleStmnt / title`

XPath

Übung

- Finden Sie alle Paragraphen in Divisions
/TEI / text / body / div / p
- Finden Sie die Personennamen innerhalb von Paragraphen
/TEI / text / body / div / p / persName
- Welches Ergebnis hat der Ausdruck?
/ TEI /teiHeader / fileDesc / titleStmnt / title
→ *das Element, das den Dokumenttitel beinhaltet*

XPath

Ergebnisse einschränken:

- um eine bestimmte Einschränkung des Abfrageergebnisses zu erlangen, können sogenannte **Prädikate** verwendet werden
- hierbei handelt es sich um zusätzliche Bedingungen, die an den Knotentest geknüpft werden

Beispiel:

```
/ TEI / text / body / div / p [2] / persName
```

Einschränkung
predicate

XPath

Übung

- Finden Sie den dritten Paragraphen in der zweiten Division

XPath

Übung

- Finden Sie den dritten Paragraphen in der zweiten Division
/ TEI / text / body / div [2]/ p [3]

XPath

Übung

- Welches Ergebnis hat der Ausdruck?

`/ TEI /teiHeader / profileDesc / particDesc / listPerson / person [2] / birth / date[2]`

XPath

Übung

- Welches Ergebnis hat der Ausdruck?

`/ TEI /teiHeader / profileDesc / particDesc / listPerson / person [2] / birth / date[2]`

→ `<date calendar="Julian">28. August 1828</date>`

XPath

Ausführliche Schreibweise für Prädikate

/ TEI / text / body / div [2]/ p [3]

entspricht

/ TEI / text / body / div [position()=2]/ p [position()= 3]

- weitere Operatoren:

für alle Datentypen:

= (gleich), != (nicht gleich)

für Zahlenwerte z.B.:

< (kleiner),
<= (kleiner oder gleich),
> (größer),
>= (größer oder gleich)

XPath

Verknüpfen von Prädikaten:

and (und) or (nicht exklusives oder)

Beispiel:

```
/ TEI / text / body / div [position()=2 ]/ p [position()= 1 or position()=3]
```

Verknüpfen von Pfadausdrücken: |

Beispiel:

```
/ TEI / text / body / div [position()=2 ]/ p [position()= 1] |
```

```
/ TEI / text / body / div [position()=2 ]/ p [position()= 3]
```

XPath

Achsen:

Achsen beschreiben die Bewegungsrichtung in der Baumstruktur eines XML-Dokuments (Vorfahren, Nachfolger und Vorgänger etc.).

Beispiel:

`/ TEI / text / body / div / descendant::persName`



Achse

XPath

Wichtige Achsen:

self:: der aktuelle Kontextknoten (.)

child:: direkte Kindelemente ()

parent:: direkter Elternknoten (..)

ancestor:: alle Vorfahren

descendant:: alle Nachkommen (//)

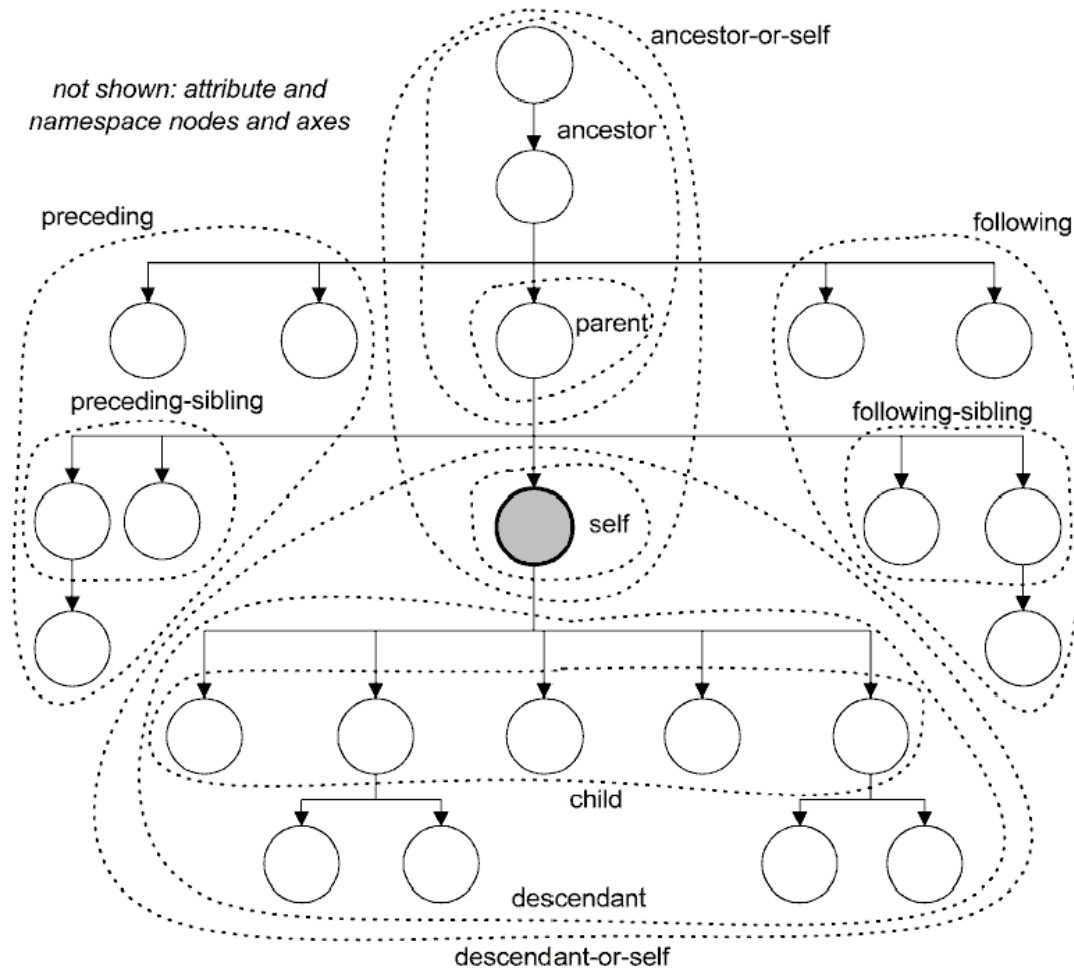
preceding:: alle Knoten vorher

following:: alle Knoten nachher

following-sibling:: alle Geschwisterknoten nachher (gemeinsamer Elternknoten)

attribute:: alle Attribute (@)

XPath



XPath Achsen, Quelle Crane Softwrights, 2001

XPath

Übung

- Finden Sie alle Personennamen in Divisions

XPath

Übung

- Finden Sie alle Personennamen in Divisions

`/TEI/text/body/div/descendant::persName` oder

`/TEI/text/body/div//persName`

XPath

Übung

- Finden Sie alle Personennamen in Divisions

`/TEI/text/body/div/descendant::persName` oder
`/TEI/text/body/div//persName`

- Finden Sie das Elternelement der zweiten Überschrift

XPath

Übung

- Finden Sie alle Personennamen in Divisions

`/TEI/text/body/div/descendant::persName` oder
`/TEI/text/body/div//persName`

- Finden Sie das Elternelement der zweiten Überschrift

`//head[2]/parent::*` oder
`//head[2]/..`

XPath

Übung

- Finden Sie alle Personennamen in Divisions

`/TEI/text/body/div/descendant::persName` oder

`/TEI/text/body/div//persName`

- Finden Sie das Elternelement der zweiten Überschrift

`//head[2]/parent::*` oder

`//head[2]/..`

- Welches Ergebnis hat folgender Ausdruck: `// div / attribute::type`

XPath

Übung

- Finden Sie alle Personennamen in Divisions

`/TEI/text/body/div/descendant::persName` oder

`/TEI/text/body/div//persName`

- Finden Sie das Elternelement der zweiten Überschrift

`//head[2]/parent::*` oder

`//head[2]/..`

- Welches Ergebnis hat folgender Ausdruck: `// div / attribute::type`
type-Attribute aller `<div>`s

XPath

Übung mit verschiedenen Knotentypen

- Finden Sie die Text-Elemente der Paragraphen in der ersten Division

XPath

Übung mit verschiedenen Knotentypen

- Finden Sie die Text-Elemente der Paragraphen in der ersten Division

```
/ TEI / text / body / div [ 1 ] / p // text()
```


XPath

Übung mit verschiedenen Knotentypen

- Finden Sie die Text-Elemente der Paragraphen in der ersten Division

```
/ TEI / text / body / div [ 1 ] / p // text()
```

- Finden Sie alle Text-Elemente innerhalb von <persName>

XPath

Übung mit verschiedenen Knotentypen

- Finden Sie die Text-Elemente der Paragraphen in der ersten Division

```
/ TEI / text / body / div [ 1 ] / p // text()
```

- Finden Sie alle Text-Elemente innerhalb von <persName>

```
//persName//text()
```

XPath

Übung mit verschiedenen Knotentypen

- Finden Sie die Text-Elemente der Paragraphen in der ersten Division

```
/ TEI / text / body / div [ 1 ] / p // text()
```

- Finden Sie alle Text-Elemente innerhalb von <persName>

```
//persName//text()
```

- Finden Sie alle xml:ids innerhalb von <person>

XPath

Übung mit verschiedenen Knotentypen

- Finden Sie die Text-Elemente der Paragraphen in der ersten Division

```
/ TEI / text / body / div [ 1 ] / p // text()
```

- Finden Sie alle Text-Elemente innerhalb von <persName>

```
//persName//text()
```

- Finden Sie alle xml:ids innerhalb von <person>

```
//person/@xml:id
```

XPath

Beispiele für die Verwendung von XPath-Funktionen:

```
/ TEI / text / body // head [starts-with(text(), 'Herz') ]
```

```
/ TEI / text / body // head / string-length()
```

```
/TEI/text/body//p[last()]
```

```
//p[ exists(./term)]
```

```
/ TEI / text / body //p [ not( exists(./ q ) ) ]
```

```
//p[1]/term/substring(text(), 1,6)
```

```
exists( //head [ . / parent::body ] )
```

```
/ TEI / text / body // div [ not( exists(./ head ) ) ]
```

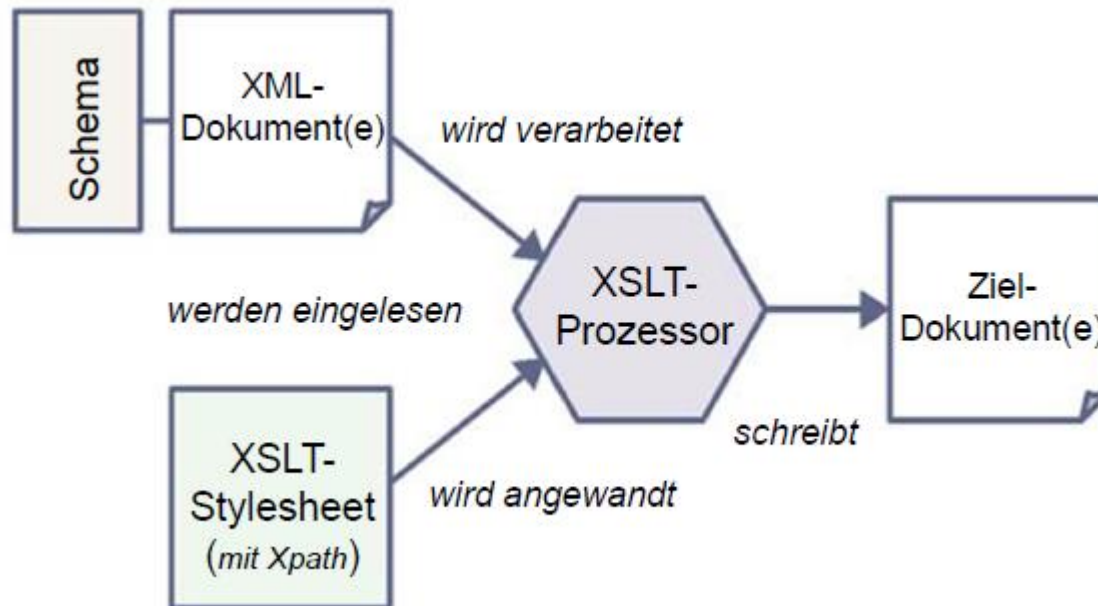
Links

<https://www.w3.org/TR/xpath-31/>

https://www.w3schools.com/xml/xsl_functions.asp

XSLT

- Transformationssprache für XML-Dokumente



XSLT – Wichtige Elemente

`<xsl:stylesheet>`

Wurzelement

`<xsl:output method="html">`

gibt an, wie das Ergebnisdokument ausgegeben werden soll (mit verschiedenen weiteren Attributen wie z.B. "xml", "text")

`<xsl:template match="/">`

eine „Schablone“, trifft hier auf die Wurzel ("/") des XML-Dokuments

`<xsl:template match="XPath">`

eine Schablone

XSLT – Wichtige Elemente

`<xsl:apply-templates/>`

wende in diesem Kontext weitere Schablonen an

`<xsl:value-of select="XPath"/>`

lese etwas aus dem XML-Dokument aus und
schreibe es in das Ergebnisdokument

`<xsl:text>`

schreibt Textdaten in das Ergebnisdokument

`<xsl:for-each select="XPath">`

wende die folgende Anweisung für jedes Element
einer Knotenmenge an („Schleife“)

XSLT – Wichtige Elemente

`<xsl:if test="XPath">`

führe die folgenden Anweisungen nur aus, wenn die Bedingung in `test` erfüllt ist

```
<xsl:choose>  
  <xsl:when test="XPath"> </xsl:when>  
  <xsl:otherwise> </xsl:otherwise>  
</xsl:choose>
```

Unterscheidung nach verschiedenen Bedingungen, die durch die Elemente `<xsl:when test="XPath">` und `<xsl:otherwise>` beschrieben werden

XSLT – Wichtige Elemente

Häufige Fehlerquellen

- der aktuelle Kontextknoten bei der Verarbeitung ist ein anderer, als man vermutet (keine Ausgabe!)
- es existieren vordefinierte Templates: Textinhalte werden immer ausgegeben (außer, man unterdrückt das ausdrücklich)
- die Elemente im Ausgangsdokument gehören einem Namensraum an, dieser wird aber in den XPath-Ausdrücken nicht berücksichtigt (keine Ausgabe!)

Links

https://www.w3schools.com/xml/xsl_transformation.asp

<https://wiki.selfhtml.org/wiki/HTML/Textstrukturierung>

<trailer>Vielen Dank für Ihre Aufmerksamkeit!</trailer>