Winter School 2024

Bergische Universität Wuppertal

20.03.2024

XML als Datenstruktur

XPath

Einführung und Übungen

Patrick Sahle

Foto: Nadine Sutor

BERGISCHE UNIVERSITÄT WUPPERTAL

DH@BUW

Dokument Text Edition
Graduiertenkolleg 2196

IZ ED
Interdisziplinäres Zentrum für Editions- und Dokumentwissenschaft

Wer hat Spaß an purer Logik?

Übersicht

- Wozu XPath?
- XPath als Standard
- Bäume, Hierarchien, Schachteln
- Konzepte und Grundbausteine
- Üben, Üben, Üben

Was ist XPath?

- "XPath is a language for addressing parts of an XML document” (XPath Specifications)
- XPath dient der Navigation in XML-Dokumenten und der Erzeugung von “Rückgaben”

- XPath wird vor allem in anderen X-Technologien verwandt: XSLT, XQuery

Was ist XPath?

- XPath ist ein W3C-Standard
- XPath 1.0 – 1999
- XPath 2.0 – 2010
- XPath 3.0 – 2014
- XPath 3.1 – 2017
- Unterstützung durch andere Technologien?
  - caveat (Python et al.)
- Was davon brauchen Sie?
  - die 10-90-Regel: Man braucht wenig um vieles zu schaffen
  - Was sind die Unterschiede und Entwicklungen der Versionen
    - neue Funktionen
    - Orientierung an Datentypen
    - andere Syntax (Operatoren)

Why should I care? Warum brauche ich XPath?

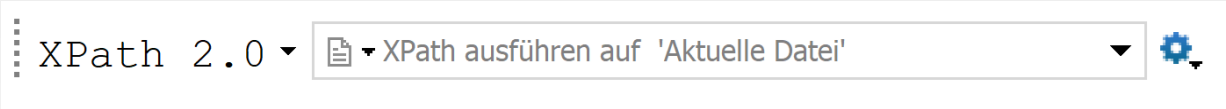
- XPath wird vor allem in anderen X-Technologien verwandt: XSLT, Xquery
  - XML-Editoren
- Selbst wenn Sie diese Technologien nicht selbst einsetzen, wollen Sie Sich in Daten zurechtfinden, Dinge finden, Sachen prüfen, Kennzahlen ermitteln etc. XPath verschafft Ihnen den Durchblick durch die Daten

Wo findet man XPath?

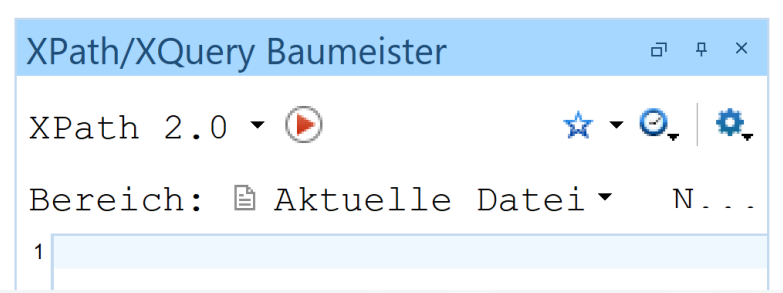
- Einfacher Einstieg: w3schools Tutorial
- Oder Sie lassen es sich von chatGPT erklären (?!)
- XPath für Geisteswissenschaftler von David Birnbaum
- Kap. 4 im Buch *XQuery for Humanists* (2020) (bei Anna)
- Ein XPath cheatsheet von Rico Sta. Cruz
- XPath-Abschnitt in der IDE-XML-Kurzreferenz
- Einfache Referenz: w3schools (nur bis 2.0!)
- Vollständige Referenz bei Maxtoroq

Einfacher Einstieg

Oben links im Editor oXygen … - der XPath-Evaluator

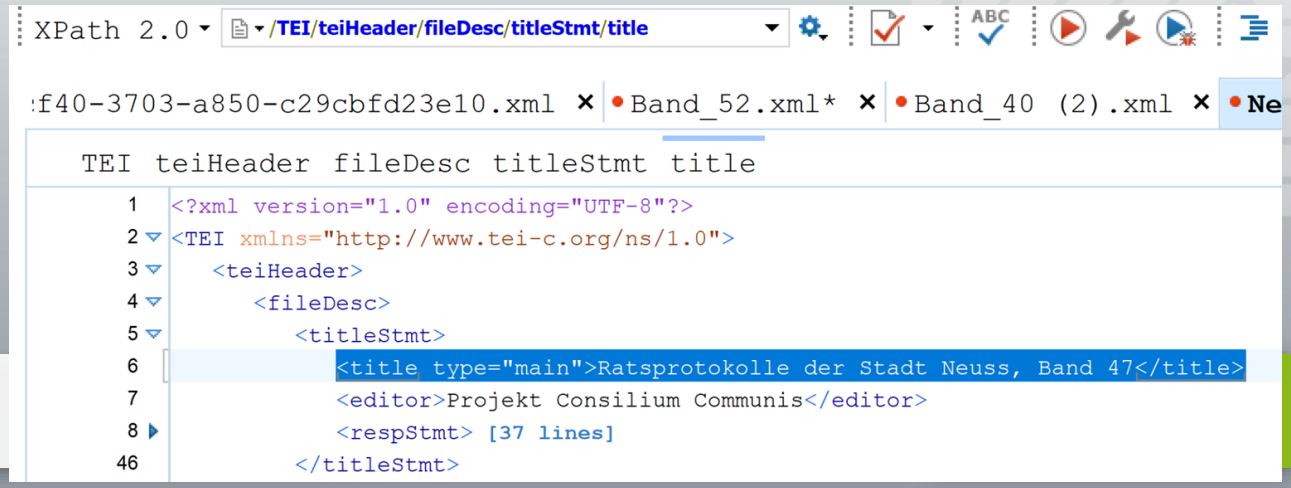


Oder in einem eigenen Fenster (für längere Pfadausdrücke) …



Ein (intuitiver?) Pfad: **/TEI/teiHeader/fileDesc/titleStmt/title**

Ein anderer Pfad: **//title**



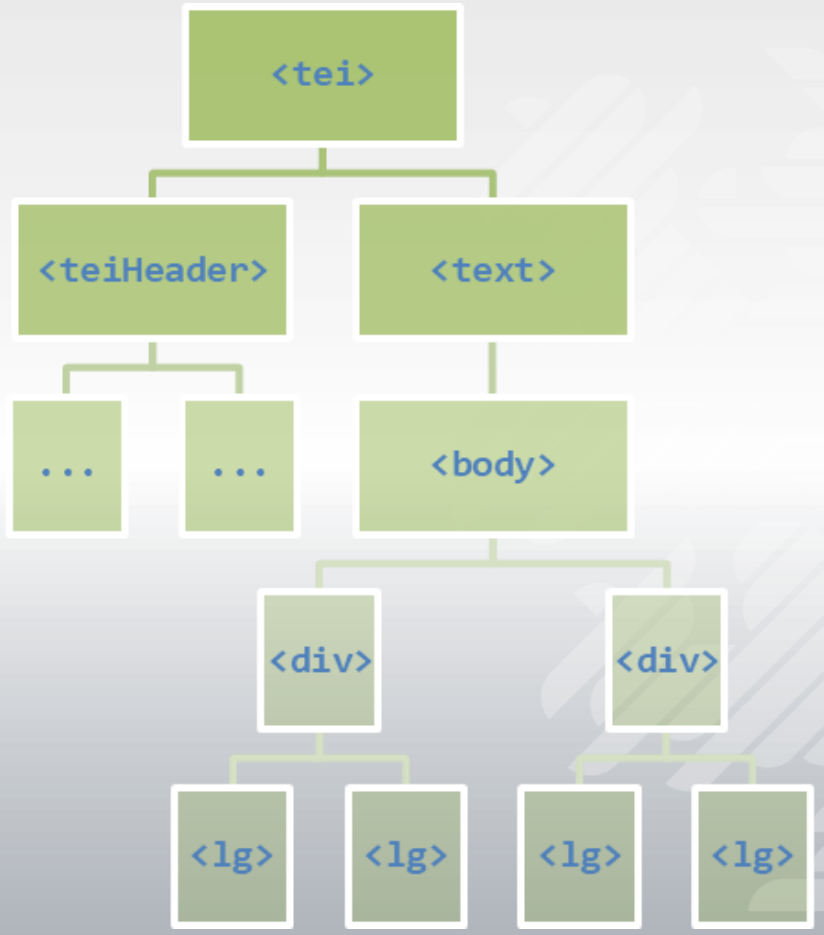
Sie haben es schon gelernt:

- Text ist sequentiell
- Tags + Inhalt = Elemente
- Elemente in Elementen = Verschachtelung
- Ein äußerstes Element, das alles umschließt
  = Eine Hierarchie!
- Ein „Baum“?

XML als “Baum”

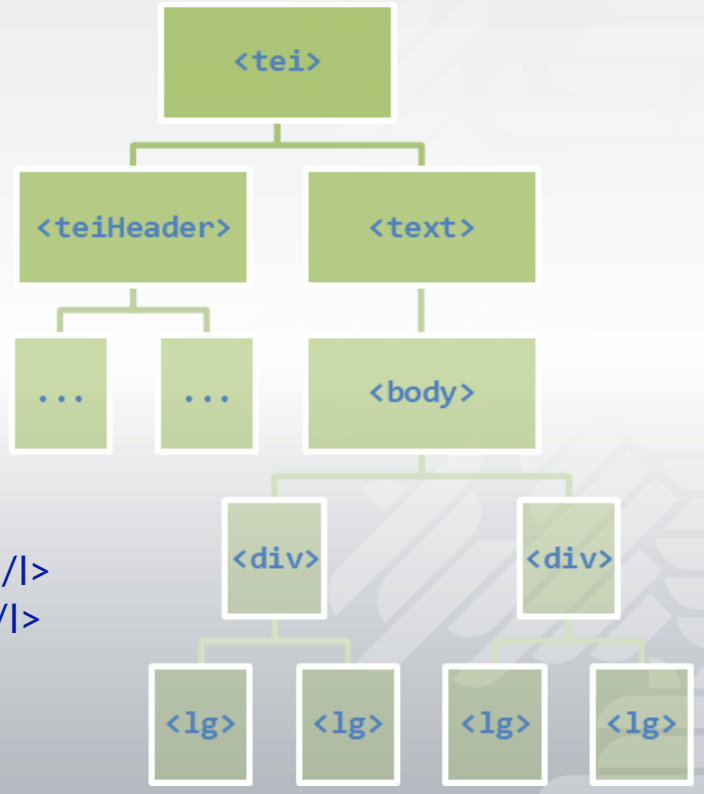
Grafik von Pi xeiKit, http://www.pixelkit.com , Liz en z : cc BY 3.0

XML als “Baum”



XML als “Baum”

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
 <teiHeader>...</teiHeader>
 <text>
   <body>
     <div type="sonnet">
       <head>Sonnet 18</head>
       <lg type="quatrain">
         <l n="1">Shall I compare thee to a summer's day?</l>
         <l n="2">Thou art more lovely and more temperate:</l>
          ...
       </lg>
        ...
       <lg type = "couplet">
         <l n="13">So long as men can breathe or eyes can see,</l>
         <l n="14">So long lives this and this gives life to thee. </l>
       </lg>
     </div>
   </body>
 </text>
</TEI>
```



Wichtige Begriffe (kennen Sie schon)

- Elemente / Attribute / Knoten
- Eltern – Kinder
- Vorfahren – Nachfahren
- Geschwister
- Wurzelknoten, Dokumentknoten

Knoten? Es gibt (nur der Vollständigkeit halber) verschiedene Typen

- Dokumentknoten
- Wurzelknoten
- Elementknoten
- Attributknoten
- Textknoten

- Kommentarknoten

Konzepte im Hintergrund: Bewegung im Baum

Lokalisierungsschritte

„gehe von hier nach da“

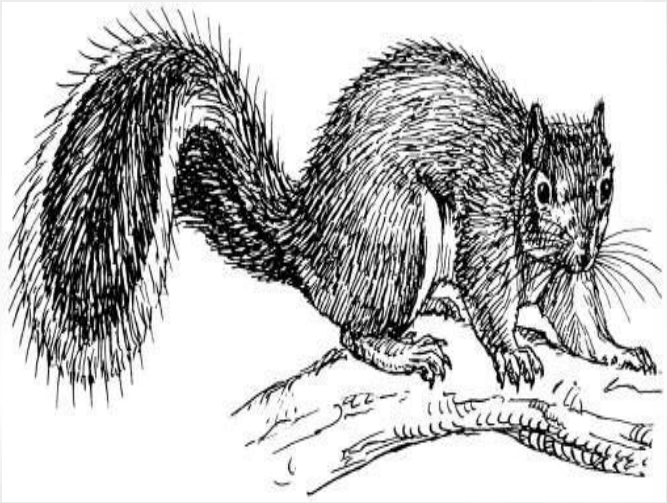
Knotentests

„bist Du der, den ich suche?“

Lokalisierungspfade

[Schritt]/[Schritt]/[Schritt]

/TEI/text/body/div



„Kontext“

absolute Pfade (vom Dokument ausgehend)

vs. relative Pfade (vom aktuellen Kontext ausgehend)

Vertikale Achsen

an den zwei Doppelpunkten könnt Ihr sie erkennen

self::

child:: descendant:: descendant-or-self::

parent:: ancestor:: ancestor-or-self::

Horizontale Achsen

following:: following-or-self:: following-sibling::

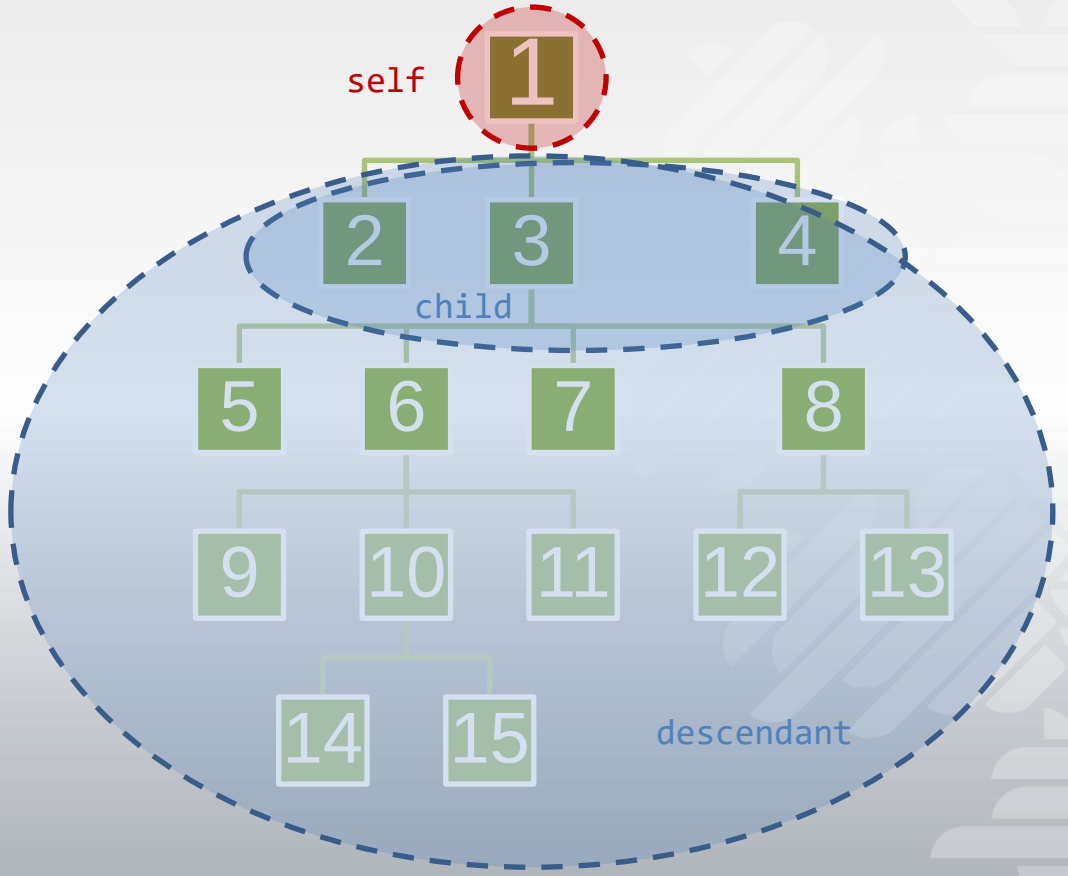
preceding:: preceding-or-self:: preceding-sibling::

Bewegung im Baum: Achsen

Selbst
`self`

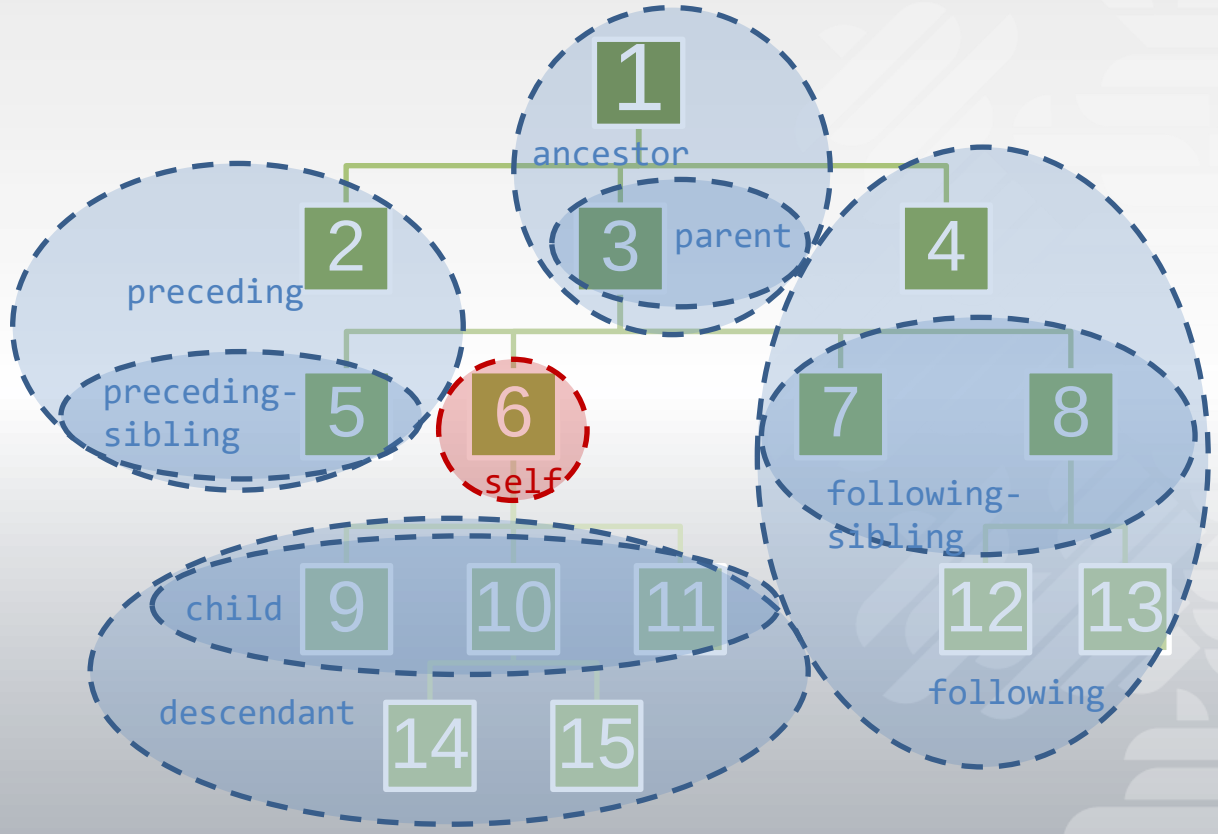
Eltern / Kind
`parent` / `child`

Vorfahren / Nachfahren
`ancestor` / `descendant`



Bewegung im Baum: Achsen

Selbst
`self`
Eltern / Kind
`parent / child`
Vorfahren / Nachfahren
`ancestor  descendant`
Geschwister
`preceding-sibling`
`following-sibling`
Sequenz statt Baum
`preceding`
`following`



Achsen: abgekürzte Syntax

Wichtig!

self::	.
child::	(nichts) Elementname
parent::	..
descendant-or-self::	//Elementname
attribute::	@Attributname

Knotentest, beliebiger Elementname *

Beispiel: /TEI/text/body/div//person/@id

Lokalisierungsschritte

Prinzip: Achse + Knotentest + **Prädikat**

Syntax: achse::knotentest**[Prädikat]**

Wichtig!

Beispiel

Gib mir die Person 175

Gib mir die Person, die im Attribut “key” den Wert 175 hat

//body/descendant::rs[@key=‘P175’]
//rs[@key=‘P175’]

Was kommt zurück?
Vier Elemente!

Der Grundbaukasten für XPath-Ausdrücke

- Verkettete Lokalisierungsschritte /…/…/
- Bedingungen, Prädikate [……]
- Klammern, Schachtelung (…(…))
- Mehrere Pfade |
- Operatoren

  and or | = != < > + - * div (etc.)

  (XPath kann Logik und auch rechnen)

  (XPath kann auch for-Schleifen und for-let-return)
- Funktionen

  Funktionsname(*Argument*, *Argument, …*)
- Syntax: 'Strings' in Anführungszeichen! Zahlen nicht!

XPath-Ausdrücke geben etwas zurück …

Wichtig!

Der letzte bzw. äußerste Schritt bestimmt, **was** ein XPath-Ausdruck zurückgibt!

Pfade werden von vorne nach hinten abgearbeitet
Klammern werden von innen nach außen aufgelöst

Beispiel

```
//body/descendant::rs[@key='P175']
```

XPath-Ausdrücke: Rückgaben

XPath-Ausdrücke ergeben Rückgaben verschiedenen Typs:

- Knoten (Knoten „mit alles“)
- Knotenmengen (sets)
- Zahlen
- Strings (Zeichenketten)
- Wahrheitswerte / Boolean (true | false)
- Sequenzen (Listen von Dingen, flach)

XPath-Ausdrücke: Rückgaben

(… gut zu wissen …)

… wenn ein Element „angesteuert“ wird, dann wird schon per default sein Textinhalt ausgegeben. Will man das explizit verlangen, kann man die Funktion text() verwenden.

XPath-Ausdrücke können **Funktionen** enthalten

- Funktionen können mit ihrem Namen aufgerufen werden.
- Funktionen bestehen aus Ihrem Namen und runden Klammern:
  funktion()
- Manche Funktionen erwarten in der Klammer die Übergabe von "etwas („Parameter“, manche kann man weglassen (optional))
  - Das kann ein Knoten sein, ein Knotensatz, ein String, eine Zahl …
- Die Übergaben sind durch Kommata getrennt
  - *funktion*(*parameter*,*parameter*)
- Funktionen geben dann etwas zurück
  - das kann ein Wahrheitswert sein, eine Zahl, ein String, eine Sequenz …

Funktionen: Ein erstes Beispiel

- Die Funktion **contains(string,string)**
  - prüft, ob ein Element oder String (das erste Argument) einen anderen String (das zweite Argument) enthält
  - liefert einen Wahrheitswert zurück
  - Auf Deutsch: Enthält der erste String den zweiten? Wahr oder falsch? True / False? In XPath-intern: true(), false()
- contains('Schnecke','ecke')
  - Deutsch: Enthält der String Schnecke den String ecke?
  - Rückgabe: true
- //forename[contains(.,‘Patrick')]
  - Deutsch: Gibt es in meinem Baum ein Element forename, das den String Patrick enthält?
  - Rückgabe: Alle Elemente forename, für die das Prädikat "true" zurückgibt (Knotenset)

Funktionen

Zum Nachschlagen, zum Lernen, zur Inspiration

- Einfache Liste (bis XPath 2.0): https://www.w3schools.com/xml/xsl_functions.asp
- Vollständige Liste: https://maxtoroq.github.io/xpath-ref/
- Lange Liste (mit XPath 3.1): https://www.altova.com/xpath-xquery-reference
- Die autoritative Referenz: https://www.w3.org/TR/xpath-functions-31/

- Woher weiß ich, welche Funktionen es gibt?
  1. Man muss eine Funktionsliste gesehen haben, um zu wissen, was es alles gibt
  2. Zu meinem Problem wird es bestimmt eine Funktion geben, die suche ich dann …

Einige Funktionen I

Gemeinsames Raten: Was tut es? Was gibt es zurück?

count(nodeset)

- zählt etwas, erwartet eine Sequenz oder ein Knotenset
- liefert eine Zahl zurück

position()

- gibt die Position eines Knotens an, liefert eine Zahl zurück
- Abgekürzte Syntax: *//person[position()=11] == //person[11]*

string-length(string)

- zählt die Länge eines Strings (in Zeichen)
- liefert eine Zahl zurück

Einige Funktionen II

starts-with(string, string)

- prüft, ob ein String mit einem anderen String beginnt, liefert einen Wahrheitswert zurück
- Starts-with('Patrick‘,'P') → true

not(boolean)

- dreht einen Wahrheitswert um, liefert einen Wahrheitswert
- not(1 > 2) → true

max(sequence of numbers) ähnlich: min(), sum(), avg()

- ermittelt den maximalen Wert aus einer Reihe von Werte, liefert eine Zahl zurück
- max(//preis) → *eine Zahl*

distinct-values(sequence of strings)

- gibt eine Sequenz von (unterschiedlichen) Werten zurück
- distinct-values(//vorname) → eine Sequenz unterschiedlicher Strings

Einige Funktionen III

- substring(string,start,length)
- matches(string, pattern)
  → reguläre Ausdrücke!
- tokenize(string, pattern)
- string()
- name()
- not(argument)
- number(string), string(number)
- doc(URI)

Einige Funktionen IV

Nur der “Vollständigkeit” halber …

- concat(string, string, …) Kurzschreibweise string || string
- translate(string1,string2,string3)
  Converts string1 by replacing the characters in string2 with the characters in string3
- → replace($eingabestring? $reg-ex, $ersatzstring?, $flags?)
- sum(arg,arg,…)
- last()
- current-date()
- xs:date()

Übungen

1. Legen Sie sich den Foliensatz bereit (zum nachschlagen)
2. Neuss_Bd_47.xml in oXygen öffnen
3. Ein Gefühl für die Daten entwickeln
4. Den XPath-Evaluator benutzen
5. Üben heißt Übersetzen (Deutsch-XPath / XPath-Deutsch)
   Man kann sich schrittweise an das Ergebnis herantasten! Man kann auch mit Pseudocode anfangen …
6. Man muss eine Idee haben, wie die logischen Schritte sind und welche Mittel (Pfade, Bedingungen, Funktionen) man einsetzen sollte. Die Tutor*innen halten Tipps bereit! (Manche Aufgaben sind echt knifflig! Ggf. Überspringen.)
7. Was kann grundlegend schief gehen? Namesräume! Nicht wohlgeformt.
8. Fragen Sie bitte nach einem Tipp, wenn nötig!

Übungen

Wer ist der Herausgeber des Dokuments?

/TEI/teiHeader/fileDesc/publicationStmt/publisher

Gib mir alle Datumsangaben in der Transkription zurück

/TEI/text/body//date

tut es auch //date (?)

Funny, aber es muss ja sein: Gib den String “Hallo Welt!” zurück

string(‘Hallo Welt!’) concat(‘Hallo ’, ‘Welt!”)

Wie heißen die Unterbereiche der file description im TEI-Headers?

//teiHeader/fileDesc/*/node-name()

Übungen

- Gib mir die einzelnen Seiten des Faksimiles (Knotenset als Rückgabe)
  - //surface
- wie breit sind die Bilder der Seiten? (→ eine Menge an Zahlen)
  - //surface/graphic/@width
- wie breit ist die erste Textregion in Faksimile 7?
  - //facsimile[@xml:id='facs_7']//zone[@rendition='TextRegion'][1]/(@lrx - @ulx)
- Was ist der Klarname des Bearbeiters mit dem Kürzel JeCl
  - //respStmt/persName[@key='JeCl']
- Wer ist der 7. Bearbeiter in der Lister der Bearbeiter?
  - //respStmt/persName[7]
- (schwieriger) Wer ist der vorletzte Bearbeiter in der Liste (count, position(), rechnen)
- Ich will die Nachnamen der Bearbeiter für die spätere Anzeige in Großbuchstaben haben (Tipp: google, gpt)
- Gib mir die Sitzungen des Stadtrats. Wie viele sind es?
- Was sind die Tagesdaten der Sitzungen im Dezember?
- Wie viele Einträge haben die einzelnen Sitzungen? (Rückgabe: Zahlen)
- Welche Sitzung hat die meisten Einträge?
- Was ist die durchschnittliche Zahl an Einträgen pro Sitzung?
- Wer sind die verschiedenen Bearbeiter in diesem Band?

Übungen

- welche Nachnamen im Personenverzeichnis gibt es, die auf …mann enden?
- tokenize: ich will die Bearbeiter*innen in der Form “Nachname, Vorname” haben
  - //respStmt/persName/concat(tokenize(.,' ')[2],', ',tokenize(.,' ')[1])
  - Bastian: //respStmt/persName/string-join(reverse(tokenize(., ' ')), ', ')
- wie viel Prozent der Orte im Ortsregister sind wirklich Orte (Stadt)
- Wieviele Schulen finden sind im Register der Organisationen?
- count(//orgName[contains(lower-case(.), 'schule')])
- “Erft” dürfte meistens eine Ortsbezeichnung sein. Ist vielleicht irgendwo vergessen worden, Erft auszuzeichnen?
  - //*[contains(string-join(./text()), 'Erft')][not(self::rs)]
- Bei manchen Personennamen werden Initialen aufgelöst (z.B. J P -> Johan Peter. Wieviele dieser Personen gibt es?
  - count(distinct-values(//rs[./expan and @type="person"]/@key))

Übungen

- Datumsangaben im Attribut: alles vollständig, sauber und unverdächtig? (verschiedene Strategien: string-length(), starts-with(.,‘19‘), kein Monat über 12?
- Wie lang sind die Pausen zwischen den Sitzungen? Was ist die längste Pause?

Vielen Dank für Ihre Aufmerksamkeit!

BERGISCHE
UNIVERSITÄT
WUPPERTAL

BERGISCHE
UNIVERSITÄT
WUPPERTAL