

The background of the slide features an abstract, textured pattern in warm tones of orange, red, and yellow at the top. Below this, a solid light gray area contains the title and author information. A thin horizontal line, matching the abstract pattern above, separates the title from the author's name.

XSL(T) en XPath

Veerle Ongenae

Overzicht Onderwerpen

- XML

- XML en namespaces

- XML Technologieën

- Structureren en datatypes: DTD en XML Schema
 - Presentatie: CSS en XSL
 - Manipulatie (o.a. transformatie): XPath, DOM en SAX
 - Diverse

Presentatie

- Scheiding van presentatie en inhoud
 - XML: data-inhoud
 - Presentatietechnologieën
- Voorbeelden
 - CSS (Cascading Style Sheets)
 - XSL en XSL-FO (Extensible Stylesheet Language - Formatting Objects)
 - XForms

CSS

- Enkel beschrijving opmaak van tags: kleur, lettertype, positionering, ...
- Binden aan XML-document

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="boeken.css"?>
```
- Voorbeelden
 - boekencss.xml
 - boeken.css

Syntax CSS

naamTag1, ..., naamtagN {eig1: waarde1;
eig2: waarde2 ... }

XSL

- Opvolger en uitbreiden mogelijkheden CSS
 - XML → HTML
 - Filteren en sorteren elementen
 - Nieuwe elementen toevoegen
 - Verwijderen of onzichtbaar maken vanelementen
- eXtensible Style Language

XSL

- XML-syntax
- Genereert uitvoer op basis van een XML-document
- Basisidee: templates (sjablonen)
 - Een template beschrijft welke acties (=genereren uitvoer) voor een bepaald element uitgevoerd moeten worden
- Kan HTML genereren

XSL

- Binden aan XML-document

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

```
<?xml-stylesheet type="text/xsl" href="boeken.xsl"?>
```

- Structuur XSL

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsl:stylesheet version="1.0"
```

```
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

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

```
  ... <!-- beschrijven van templates -->
```

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

XSL

■ Template

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

```
  <ul>
```

```
    <xsl:apply-templates/>
```

```
  </ul>
```

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

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

```
  <i>ISBN</i><xsl:text> </xsl:text>
```

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

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

Voorbeeld

- boekenxsl.xml
- boeken.xsl

XSL-elementen

- Basiselement

 - xsl:stylesheet**

- Onmiddellijke kinderen basiselement

 - xsl:output**

 - xsl:template**

 - ...

Onmiddellijke kinderen basiselement

- `<xsl:output method="uitvoertype"/>`
 - Bepaalt type van het resultaat van de transformatie
- `<xsl:template match="patroon"> ...</xsl:template>`
 - Beschrijving uit te voeren opdrachten
 - “patroon” bepaalt voor welke elementen
- ...

XSL-elementen

■ Instructie-elementen

xsl:value-of

xsl:for-each

xsl:sort

xsl:if

xsl:choose

xsl:apply-templates

xsl:text

xsl:number

...

xsl:value-of

- `<xsl:value-of select="uitdrukking"/>`
- “uitdrukking” bepaalt element of knoop
- Element
 - Inhoud wordt uitgeschreven
- Attribuut
 - Waarde wordt uitgeschreven
- Voorbeeld

```
<xsl:template match="title">  
  <i><xsl:value-of select="."/></i>  
</xsl:template>
```

xsl:for-each

- `<xsl:for-each select="uitdrukking"> ...
</xsl:for-each>`
- Een herhaling of lus
- “uitdrukking” bepaalt een aantal elementen
- Voor elk element worden de opdrachten in het xsl:for-each element uitgevoerd

xsl:for-each Voorbeeld

```
<xsl:template match="author">
  <xsl:for-each select="firstname">
    <xsl:apply-templates select="."/>
    <xsl:text> </xsl:text>
  </xsl:for-each>
  ...
</xsl:template>
```

xsl:sort

- Sorteert de uitvoer van een xsl:for-each of xsl:apply-templates element

`<xsl:sort select="uitdrukking" order="volgorde" data-type="type" />`

- “uitdrukking” bepaalt het element of attribuut waarop gesorteerd wordt
- “volgorde” bepaalt de volgorde
 - ascending of descending (standaard ascending)
- “type” bepaalt of er getallen of strings gesorteerd worden
 - text of number (standaard text)

xsl:sort Voorbeeld

```
<xsl:for-each select="catalog/cd">
  <xsl:sort select="artist"/>
  <tr><td>
    <xsl:value-of select="title"/>
  </td><td>
    <xsl:value-of select="artist"/>
  </td></tr>
</xsl:for-each>
```

xsl:if

- ALS voorwaarde waar DOE

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

```
  opdrachten
```

```
</xsl:if>
```

- “voorwaarde”

- Waar → opdrachten uitgevoerd
- Niet vervuld → niets

xsl:if Voorbeeld

```
<xsl:for-each select="catalog/cd">
  <xsl:if test="price > 10">
    <tr><td>
      <xsl:value-of select="title"/>
    </td><td>
      <xsl:value-of select="artist"/>
    </td></tr>
  </xsl:if>
</xsl:for-each>
```

xsl:choose

- Meerdere gevallen
- Per geval andere opdrachten

```
<xsl:choose>
```

```
  <xsl:when test="geval1">opdrachten</xsl:when>
```

```
  <xsl:when test="geval2">opdrachten</xsl:when>
```

```
  ...
```

```
  <xsl:when test="gevalN">opdrachten</xsl:when>
```

```
  <xsl:otherwise>opdrachten</xsl:otherwise>
```

```
</xsl:choose>
```

xsl:apply-templates

- `<xsl:apply-templates/>`
 - Pas de template toe van alle kinderen van het huidige element
- `<xsl:apply-templates select="uitdrukking">`
 - Past de template toe van de elementen bepaald door "uitdrukking"
- Wordt voor een bepaald element geen template voorzien
 - Inhoud element wordt uitgeschreven

xsl:apply-templates Voorbeeld

```
<xsl:template match="books">
  <ul>
    <xsl:apply-templates/>
  </ul>
</xsl:template>
...
<xsl:for-each select="author">
  <xsl:apply-templates select="."/>
  <xsl:text>, </xsl:text>
</xsl:for-each>
```

xsl:text

- Schrijft een stukje tekst naar de uitvoer
- Handig voor spaties

xsl:variable

- Verbindt een naam met een waarde
- Geen echte variabele → waarde kan niet aangepast worden

- Syntax

`<xsl:variable name="naam" select="waarde"/>`

- name: naam
- select: waarde, een uitdrukking

`<xsl:variable name="naam">waarde</xsl:variable>`

- Kan elders gebruik worden
 - \$naam

Voorbeeld

- boekenMetVariabele.xsl

xsl:param en xsl:with-param

- Gebruik
 - Parameter meegeven aan template
- xsl:param
 - Parameter declareren in template
- xsl:with-param
 - Parameter meegeven aan template

xsl:param en xsl:with-param

■ Syntax

■ xsl:param

```
<xsl:template ...>
```

```
...
```

```
<xsl:param name="naam" select="waarde"/>
```

```
...
```

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

- name: naam → elders \$naam

- select: standaardwaarde

xsl:param en xsl:with-param

■ Syntax

■ xsl:with-param

```
<xsl:apply-templates ...>
```

```
  <xsl:with-param name="naam"  
    select="waarde"/>
```

```
</xsl:apply-templates>
```

- name: naam parameter
- select: waarde parameter, wordt meegegeven aan opgeroepen template

Voorbeeld

- servers.dtd
- serversMetBeelden.xml
- serversMetBeelden.xsl

Attributen van XSL-elementen

- Uitdrukkingen in XPath
- Voorbeelden

•

**count(books/book)
title**

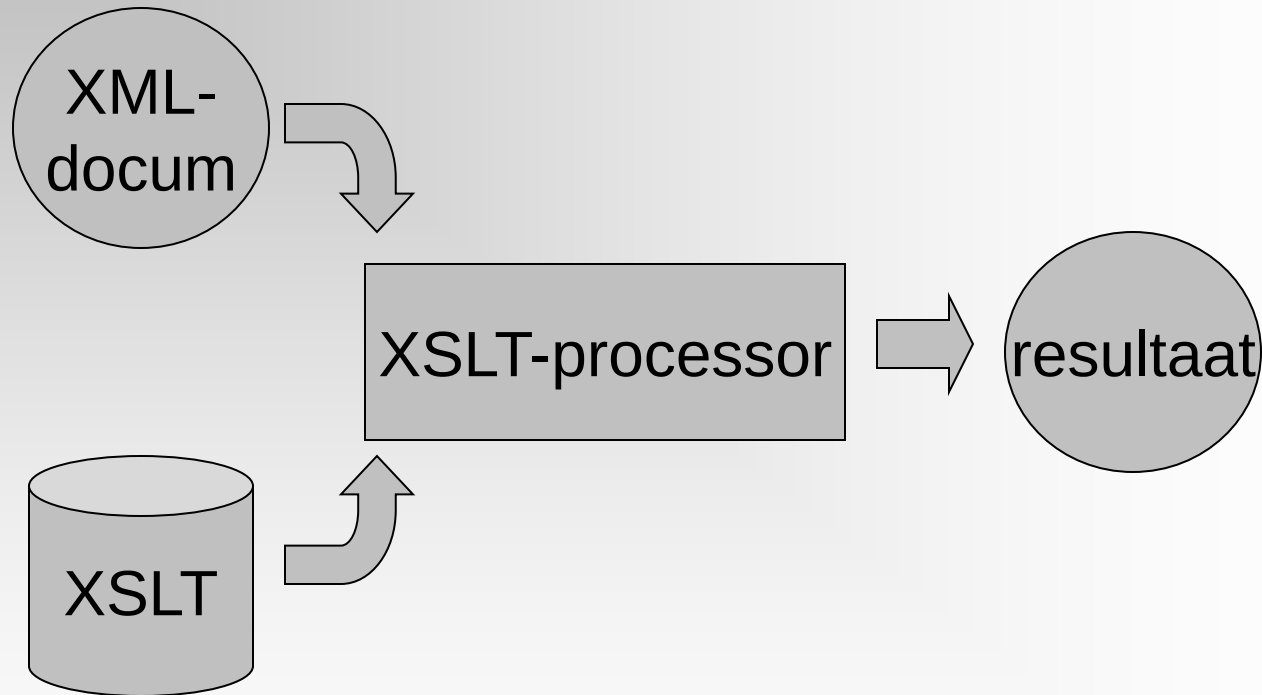
Overzicht Onderwerpen XML

- XML
 - XML en namespaces
- XML Technologieën
 - Structureren en datatypes: DTD en XML Schema
 - Presentatie: CSS en XSL
 - Manipulatie (o.a. transformatie): XSLT, XPath, DOM en SAX
 - Diverse

XSLT

- XSL: opvolger van CSS, presentatie XML-documenten
- Ook bruikbaar voor transformatie van XML-documenten naar andere XML-formaten of naar andere formaten
- XSLT-document beschrijft de translatie
- Voorbeeld
 - Docbook naar LaTeX
 - XML naar XSL-FO

Principe transformatie



XSLT

- Transformatie wordt beschreven aan de hand van templates
- Gebruikt XPath om stukken XML te selecteren
- XSLT-processor leest XML-document, XSLT-document en genereert een outputdocument
- XSLT-processors
 - SAXON
 - Xalan (maakt deel uit van java 1.4)

XSL-FO

- XML-taal om de layout van tekstopagina's te beschrijven (XML-formaat voor drukwerk)
- XSL-FO wordt meestal gegenereerd door een XSLT-processor die een XML-document omzet aan de hand van een XSLT
- XSL-FO wordt vervolgens om gezet naar een ander formaat bv. PDF

Informatie

- <http://www.w3c.org/Style/XSL/>
- <http://www.w3schools.com/xsl/default.asp>

XPath

- Wat is XPath?
- Identificatie van knopen: paden
- Uitdrukkingen

XPath

- Taal om stukken van een XML-document te identificeren
- XSL(T) gebruikt XPath-uitdrukkingen om stukken XML te selecteren
- Beschouwt een XML-document als een boomstructuur met knopen
- Een XPath-uitdrukking heeft als resultaat
 - Eén of meerdere knopen
 - Getallen, booleans of strings

XPath: Knopen

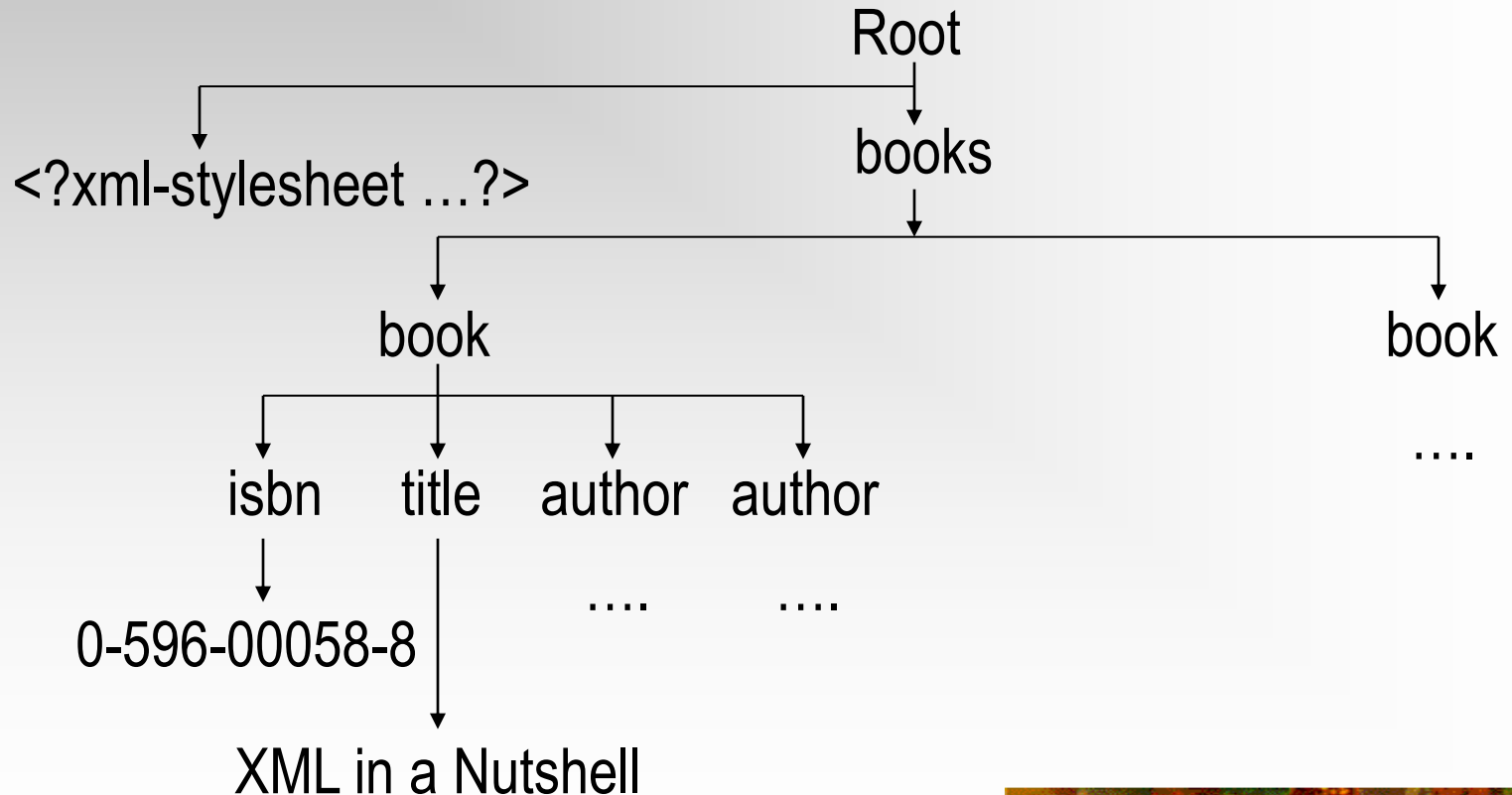
- Mogelijke knopen
 - Basisknoop (root node)
 - Elementen
 - Tekst
 - Attributen
 - Commentaar
 - Processing Instructions
 - Namespace

XPath: Knopen

- Geen knopen
 - CDATA
 - Entiteiten
 - DTD

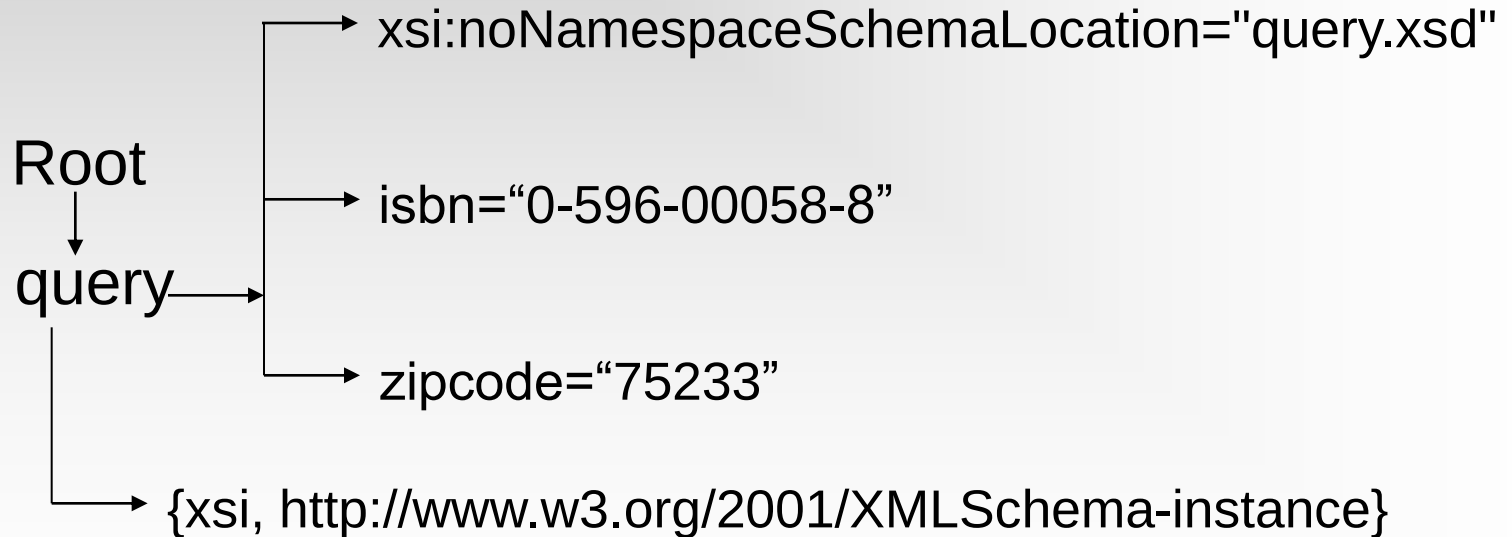
Voorbeeld XPath-boom

■ boekenxsl.xml



Voorbeeld XPath-boom

■ queryxsd.xml



XPath

- Wat is XPath?
- Identificatie van knopen: paden
- Uitdrukkingen

XPath: Location Path (Pad)

- Identificeert één of meerdere knopen
- Een pad bestaat uit verschillende stappen (location steps) verbonden met een /
- Elke stap wordt uitgevoerd in een bepaalde context (een bepaalde knoop - context node)
- Vergelijkbaar met een bestandssysteem

XPath: Padstructuur

■ Structuur pad

- Basispad
- Kindelement-stap
- Attribuutstap
- Stap naar commentaar, tekst of Processing Instructions

■ Andere padelementen

- Jokertekens
- Meerdere mogelijkheden
- Speciale tekens

■ Voorwaarden

46 ■ Lange padnamen

XPath: mogelijke paden

- Basispad (root location path): /
 - Root van de XPath-boom
- Kind-element-stap (child element location step)
 - Alle elementknopen die kinderen zijn van de huidige knoop (context)
 - Voorbeelden
 - book (afhankelijk van de context)
 - /book
 - /books/book

XPath en XSLT: Kind-element-stap

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

```
...
```

```
<xsl:apply-templates select="name" />
```

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

XPath: mogelijke paden

- Attribuut-stap (attribute location step)
 - Selecteert een attribuut van een element (de huidige knoop)
 - @ + naam attribuut
 - Voorbeelden
 - @zipcode
 - /query/@isbn

XPath en XSLT: Attribuut-stap

```
<xsl:template match="query">  
  <xsl:value-of select="@isbn" />  
</xsl:template>
```

XPath: mogelijke paden

- Stap naar commentaar, tekst en processing-instructions
 - `comment()`
 - `text()`
 - `processing-instruction()`
- Selecteert respectievelijk commentaar, tekst of processing-instructions in de huidige context

XPath: Stap naar commentaar, tekst en ...

■ Voorbeelden

- `processing-instruction()`
- `processing-instruction('xml-stylesheet')`
- `book/isbn/text()`

XPath: Padstructuur

- Structuur pad
 - Basispad
 - Kindelement-stap
 - Attribuutstap
 - Stap naar commentaar, tekst of Processing Instructions
- Andere padelementen
 - Jokertekens
 - Meerdere mogelijkheden
 - Speciale tekens
- Voorwaarden
- Lange padnamen

XPath: mogelijke paden - jokertekens

- Jokertekens (Wildcards)
 - *: willekeurige elementknoop
 - node(): willekeurige knoop (alle types)
 - @*: willekeurige attribuutknoop
 - Voorbeelden
 - /query/@*

XPath: mogelijke paden – meerdere mogelijkheden

- Meerdere mogelijkheden

- |

- Voorbeelden

- isbn|title

- /books/book/isbn|title

XPath: mogelijke paden – speciale tekens

■ Speciale tekens.

■ .

- De huidige knoop

■ //

- Alle elementen onder de huidige knoop
- Voorbeelden
 - //book
 - ../name

■ ..

- De ouder van de huidige knoop

XPath: Padstructuur

- Structuur pad
 - Basispad
 - Kindelement-stap
 - Attribuutstap
 - Stap naar commentaar, tekst of Processing Instructions
- Andere padelementen
 - Jokertekens
 - Meerdere mogelijkheden
 - Speciale tekens
- Voorwaarden

XPath: paden met voorwaarden

■ Voorwaarde

- Verwijdt knopen uit de selectie die niet aan de voorwaarde voldoen
- Tussen vierkante haken []
- Logische uitdrukking
- Operatoren
 - <, >, >=, <=, !=, =, or, and
 - Let op soms <

XPath: paden met voorwaarden

- Voorbeelden

- `/query[@zipcode=75233]`
- `//book[title="XML"]`
- `//name[.='Ongenae']`

- Opmerking

- Waarde van een element is de tekstinhoud van het element
- Waarde van een attribuut is de waarde van het attribuut

XPath: Padstructuur

- Structuur pad
 - Basispad
 - Kindelement-stap
 - Attribuutstap
 - Stap naar commentaar, tekst of Processing Instructions
- Andere padelementen
 - Jokertekens
 - Meerdere mogelijkheden
 - Speciale tekens
- Voorwaarden

XPath: lange padnamen

- Tot nu toe
 - Afgekorte padnamen
 - Best voor XSLT
- Elke stap
 - As (impliciet voor verkorte vorm)
 - Bepaalt deel van de boom waarin 'gekeken' wordt
 - Test om knopen te selecteren
 - Voorwaarde (optioneel)
- As en Test worden gescheiden door ::

XPath: lange padnamen

■ Voorbeeld

- `child::books/child::book/child::isbn`
- `child::query/attribute::isbn`

XPath: lange padnamen - assen

- Assen die afgekort kunnen worden
 - child
 - parent (..)
 - self (.)
 - attribute (@)
 - descendant-or-self (//)
- Merk op attributen zijn geen kinderen

XPath: lange padnamen - assen

■ Andere assen

- ancestor: alle voorouders
- ancestor-or-self: alle voorouders + huidige knoop
- namespace: alle namespaces in de huidige context
- descendant: alle nakomelingen (kinderen van kinderen en zo)

XPath: lange padnamen - assen

■ Andere assen

- following-sibling: knopen na de huidige knoop en kinderen van dezelfde ouder; attributen en namespaces hebben geen siblings
- preceding-sibling: knopen voor de huidige knoop en kinderen van dezelfde ouder; attributen en namespaces hebben geen siblings
- following: knopen na de huidige knoop (geen attributen of namespaces)
- preceding: knopen voor de huidige knoop (geen attributen of namespaces)

XPath

- Wat is XPath?
- Identificatie van knopen: paden
- Uitdrukkingen

XPath: algemene uitdrukkingen

- Paden: identificatie van één of meerdere knopen
- Andere mogelijkheden: getallen, booleans en strings
- Bv. Waarde van het select-attribuut van het element `<xsl:value-of>` in XSL

XPath: uitdrukkingen

■ Datatypes

- Getallen
- Strings
- Logische variabelen

■ Functies

- Knopen
- Strings
- Logische
- Numerieke

XPath: datatype - getallen

- Type: IEEE 64-bit coding (cfr. double in Java)
- Operatoren
 - +
 - -
 - *
 - div
 - mod
- Voorbeeld
 - `<xsl:value-of select="(@born-(@born mod 100)) div 100 + 1"/>`

XPath: datatype - strings

- Unicode (geen aanhalingstekens)
- Operatoren
 - =
 - !=

XPath: datatype - booleans

- Geen constanten true of false, wel functies true() en false()
- Resultaat van vergelijkingen
- Gebruik: vooral in voorwaarden
 - XPath
 - XSL en XSLT
- Operatoren: and or

XPath: uitdrukkingen

■ Datatypes

- Getallen
- Strings
- Logische variabelen

■ Functies

- Knopen
- Strings
- Logische
- Numerieke

XPath: functies

- Resultaat: boolean, getal, string, knopen

XPath: Knopenfuncties

- Mogelijke knopenfuncties
 - Resultaat knopen
 - Bewerkingen op knopen
- Voorbeelden
 - position()
 - last(): positie van de laatste knoop = aantal knopen in huidige knoop
 - count(arg): aantal knopen bepaald door het argument arg
 - ...

XPath: String-functies

■ Voorbeelden

- `string(...)`: converteer argument naar een string
- `starts-with(..., ...)`: controleert of het eerste argument start met het tweede
- `contains(..., ...)`: controleert of het eerste argument het tweede bevat
- `concat(..., ...)`
- ...

XPath: logische functies

- `true()`
- `false()`
- `not(...)`
- `boolean(...)`

XPath: numerieke functies

- `number(...)`
- `round(...)`
- `floor(...)`
- `ceiling(...)`
- `sum(...)`