

XSL(T) EN XPATH

Veerle Ongenae

UNIVERSITEIT GENT

FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Doel

- Basisconcepten XSLT kennen
 - Wat is XSLT?
 - Hoe werkt het?
- Structuur XSLT-document kennen en toepassen
- Basisprincipes XPath kennen
 - Wat is XPath?
 - Rol XPath in XSLT
 - Structuur van een pad
 - Verschillende assen

Industrieel Ingenieur Informatica, UGent

Overzicht XML

- XML
 - XML en namespaces
- XML Technologieën
 - Structureren en datatypes: DTD en XML Schema
 - Transformatie: XSLT

Industrieel Ingenieur Informatica, UGent

Wat is XSLT?

```

  graph LR
    XML1[XML] -- XSLT --> XML2[XML]
    XML1 -- XSLT --> HTML[HTML]
    XML1 -- XSLT --> tekst[tekst]
  
```

- eXtensible Style Language Transformations
- XML-taal
- Transformatie XML-document naar andere formaten
- Basisidee: templates (sjablonen)
 - Een template beschrijft welke acties (=genereren uitvoer) voor een bepaald element uitgevoerd moeten worden

Industrieel Ingenieur Informatica, UGent

Concept XSLT

- Binden aan XML-document


```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="boeken.xsl"?>
```
- Externe XSLT-processor

```

  graph LR
    XML((XML-document)) --> XSLTProc[XSLT-processor]
    XSLT[(XSLT)] --> XSLTProc
    XSLTProc --> Result((resultaat))
  
```

Industrieel Ingenieur Informatica, UGent

Structuur XSLT-document

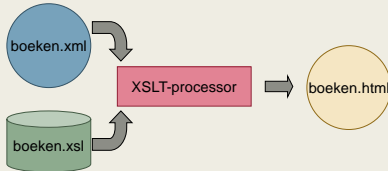
- Baselement
 - `xsl:stylesheet`
- Onmiddellijke kinderen baselement
 - `xsl:output`
 - type resultaat transformatie: xml, html, tekst, ...
 - `xsl:template`
 - uit te voeren opdrachten
 - "patroon" bepaalt voor welke elementen

```

<?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"/>
  <xsl:template match="patroon">
    ...
  </xsl:template>
</xsl:stylesheet>
  
```

Industrieel Ingenieur Informatica, UGent

Voorbeeld



Overzicht boeken

Er zijn 2 boeken in de catalogus.

- *XML in a Nutshell*, Elliotte Rusty Harold, Scott W. Means, ISBN 0-596-00058-8
- *XML, Web Services, and the Data Revolution*, Frank P. Coyle, ISBN 0-201-77641-3

Industrieel Ingenieur Informatica, UGent

7

Voorbeeld



```

<?ELEMENT books (book*)>
<?ELEMENT book (isbn, title, author+)>
<?ELEMENT isbn (#PCDATA)>
<?ELEMENT title (#PCDATA)>
<?ELEMENT author (name, initial?, firstname+)>
<?ELEMENT name (#PCDATA)>
<?ELEMENT initial (#PCDATA)>
<?ELEMENT firstname (#PCDATA)>
  
```

Overzicht boeken

Er zijn 2 boeken in de catalogus.

- *XML in a Nutshell*, Elliotte Rusty Harold, Scott W. Means, ISBN 0-596-00058-8
- *XML, Web Services, and the Data Revolution*, Frank P. Coyle, ISBN 0-201-77641-3

Industrieel Ingenieur Informatica, UGent

8

Voorbeeld – start pagina

Overzicht boeken

Er zijn 2 boeken in de catalogus.

- *XML in a Nutshell*, Elliotte Rusty Harold, Scott W. Means, ISBN 0-596-00058-8
- *XML, Web Services, and the Data Revolution*, Frank P. Coyle, ISBN 0-201-77641-3

```

<xsl:template match="/">
  <html>
  <head>
    <title>Overzicht boeken</title>
  </head>
  <body>
    <h1>Overzicht boeken</h1>
    <xsl:text>Er zijn </xsl:text>
    <xsl:value-of select="count(books/book)"/>
    <xsl:text> boeken in de catalogus.</xsl:text>
    <xsl:apply-templates/>
  </body>
</html>
</xsl:template>
  
```

xsl:text

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

```

<xsl:template match="/">
  ...
  <xsl:text>Er zijn </xsl:text>
  <xsl:value-of select="count(books/book)"/>
  <xsl:text> boeken in de catalogus.</xsl:text>
  ...
</xsl:template>
  
```

Industrieel Ingenieur Informatica, UGent

10

xsl:value-of

- <xsl:value-of select="uitdrukking"/>
 - "uitdrukking" bepaalt element, attribuut, ...
 - Element
 - Inhoud wordt uitgeschreven
 - Attribuut
 - Waarde wordt uitgeschreven

```

<xsl:template match="/">
  ...
  <xsl:text>Er zijn </xsl:text>
  <xsl:value-of select="count(books/book)"/>
  <xsl:text> boeken in de catalogus.</xsl:text>
  ...
</xsl:template>
  
```

Industrieel Ingenieur Informatica, UGent

11

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"
- Geen template → inhoud element uitschrijven

```

<xsl:template match="/">
  ...
  <xsl:apply-templates/>
  ...
</xsl:template>

<xsl:template match="books">
  <ul>
    <xsl:apply-templates/>
  </ul>
</xsl:template>
  
```

Industrieel Ingenieur Informatica, UGent

12

Voorbeeld – lijst boeken

Overzicht boeken

Er zijn 2 boeken in de catalogus.

- *XML in a Nutshell*, Elliotte Rusty Harold, Scott W. Means, ISBN 0-596-00058-8
- *XML, Web Services, and the Data Revolution*, Frank P. Coyle, ISBN 0-201-77641-3

```
<xsl:template match="books">
  <ul>
    <xsl:apply-templates/>
  </ul>
</xsl:template>

<xsl:template match="book">
  <i>
    <xsl:apply-templates select="title"/>
    <xsl:text>, </xsl:text>
    <xsl:for-each select="author">
      <xsl:apply-templates select="."/>
    </xsl:for-each>
    <xsl:text>, </xsl:text>
    <xsl:apply-templates select="isbn"/>
  </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:template match="book">
  ...
  <xsl:for-each select="author">
    <xsl:apply-templates select="."/>
    <xsl:text>, </xsl:text>
  </xsl:for-each>
  ...
</xsl:template>
```

Industrieel Ingenieur Informatica, UGent

14

Voorbeeld – isbn – titel -auteur

Overzicht boeken

Er zijn 2 boeken in de catalogus.

- *XML in a Nutshell*, Elliotte Rusty Harold, Scott W. Means, ISBN 0-596-00058-8
- *XML, Web Services, and the Data Revolution*, Frank P. Coyle, ISBN 0-201-77641-3

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

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

XSL-elementen

- Instructie-elementen
 - xsl:text
 - xsl:value-of
 - xsl:apply-templates
 - xsl:for-each
 - xsl:sort
 - xsl:if
 - xsl:choose
 - xsl:variable
 - xsl:param en xsl:with-param
 - xsl:number
 - ...

Industrieel Ingenieur Informatica, UGent

16

xsl:sort

```
<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>
```

- Sorteert de uitvoer van
 - xsl:for-each
 - xsl:apply-templates
- <xsl:sort select="uitdrukking" order="volgorde" data-type="type"/>
- "uitdrukking"
 - Waarop sorteren?
- "volgorde"
 - ascending (standaard) of descending
- "type"
 - Hoe sorteren?
 - text (standaard) of number

Industrieel Ingenieur Informatica, UGent

17

xsl:if

```
<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>
```

- ALS voorwaarde waar DOE
- <xsl:if test="voorwaarde">
- opdrachten
- </xsl:if>
- "voorwaarde"
 - Waar → opdrachten uitgevoerd
 - Niet vervuld → niets

Industrieel Ingenieur Informatica, UGent

18

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>
```

Industrieel Ingenieur Informatica, UGent

19

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
- Kan elders gebruik worden


```
$naam
```

Industrieel Ingenieur Informatica, UGent

20

Voorbeeld xsl:variabele

Overzicht boeken

Er zijn 2 boeken in de catalogus. Er zijn te weinig boeken.

- XML in a Nutshell, Elliottte Rusty Harold, Scott W. Meams, ISBN 0-596-00058-8
- XML, Web Services, and the Data Revolution, Frank P. Coyle, ISBN 0-201-77641-3

```
<xsl:template match="/">
  ...
  <h1>Overzicht boeken</h1>
  <xsl:variable name="aantal" select="count(books/book)"/>
  <xsl:variable name="min">10</xsl:variable>
  <xsl:text>Er zijn </xsl:text>
  <xsl:value-of select="$aantal"/>
  <xsl:text> boeken in de catalogus. </xsl:text>
  <xsl:if test="$aantal < $min">
    <xsl:text>Er zijn te weinig boeken.</xsl:text>
  </xsl:if>
  <xsl:apply-templates/>
  ...
</xsl:template>
```

Industrieel Ingenieur Informatica, UGent

21

xsl:param en xsl:with-param

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


```
<xsl:template ...>
  <xsl:param name="naam" select="waarde"/>
  ...
</xsl:template>
```
- xsl:with-param
 - Parameter meegeven aan template waarde param in template
 - ```
<xsl:apply-templates ...>
 <xsl:with-param name="naam" select="waarde"/>
 ...
</xsl:apply-templates>
```

Industrieel Ingenieur Informatica, UGent

22

## Voorbeeld xsl:param

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

```
<?xml version="1.0"?>
<!DOCTYPE servers SYSTEM "servers.dtd">
<servers>
 <url>http://www.mijnfoto.be</url>
 <foto bron="IMG001">Mijn huis</foto>
 <foto bron="IMG002">Kinderen</foto>
 ...
</servers>
```



```
<ELEMENT servers (server*)>
<ELEMENT server (url,foto*)>
<ELEMENT url (#PCDATA)>
<ELEMENT foto (#PCDATA)>
<!ATTLIST foto bron NMTOKEN #REQUIRED>
```

### Overzicht foto's

http://www.mijnfoto.be	
Mijn huis	http://www.mijnfoto.be/IMG001
Kinderen	http://www.mijnfoto.be/IMG002
Poizen	http://www.mijnfoto.be/IMG003
Tuin	http://www.mijnfoto.be/IMG004
Vakantie in Praag	http://www.mijnfoto.be/IMG005
Scoutskamp	http://www.mijnfoto.be/IMG006
Sneeuwpret	http://www.mijnfoto.be/IMG007

http://www.bedreigdedieren.org	
Benzaalse tijger	http://www.bedreigdedieren.org/IMG001
Panda	http://www.bedreigdedieren.org/IMG002
Koala	http://www.bedreigdedieren.org/IMG003
Wolff	http://www.bedreigdedieren.org/IMG004

## Voorbeeld xsl:param

```
<xsl:template match="/">
 <html>
 <head>
 <title>Overzicht foto's</title>
 </head>
 <body>
 <h1>Overzicht foto's</h1>
 <xsl:apply-templates select="servers/server"/>
 </body>
 </html>
</xsl:template>
```

### Overzicht foto's

http://www.mijnfoto.be	
Mijn huis	http://www.mijnfoto.be/IMG001
Kinderen	http://www.mijnfoto.be/IMG002
Poizen	http://www.mijnfoto.be/IMG003
Tuin	http://www.mijnfoto.be/IMG004
Vakantie in Praag	http://www.mijnfoto.be/IMG005
Scoutskamp	http://www.mijnfoto.be/IMG006
Sneeuwpret	http://www.mijnfoto.be/IMG007

http://www.bedreigdedieren.org	
Benzaalse tijger	http://www.bedreigdedieren.org/IMG001
Panda	http://www.bedreigdedieren.org/IMG002
Koala	http://www.bedreigdedieren.org/IMG003
Wolff	http://www.bedreigdedieren.org/IMG004

## Voorbeeld xsl:param

http://www.mijnfoto.be	
Mijn huis	http://www.mijnfoto.be/IMG001
Kinderen	http://www.mijnfoto.be/IMG002
Dozen	http://www.mijnfoto.be/IMG003
Tuin	http://www.mijnfoto.be/IMG004
Vakantie in Praag	http://www.mijnfoto.be/IMG005
Scoutskamp	http://www.mijnfoto.be/IMG006
Saferuipert	http://www.mijnfoto.be/IMG007

```
<xsl:template match="server">
 <xsl:variable name="urlhost" select="url"/>
 <table border="1">
 <tr>
 <th colspan="2">
 <xsl:value-of select="$urlhost"/>
 </th>
 </tr>
 <xsl:apply-templates select="foto">
 <xsl:with-param name="host" select="$urlhost"/>
 </xsl:apply-templates>
 </table>
</xsl:template>
```

## Voorbeeld

http://www.mijnfoto.be	
Mijn huis	http://www.mijnfoto.be/IMG001
Kinderen	http://www.mijnfoto.be/IMG002
Dozen	http://www.mijnfoto.be/IMG003
Tuin	http://www.mijnfoto.be/IMG004
Vakantie in Praag	http://www.mijnfoto.be/IMG005
Scoutskamp	http://www.mijnfoto.be/IMG006
Saferuipert	http://www.mijnfoto.be/IMG007

```
<xsl:template match="foto">
 <xsl:param name="host"/>
 <xsl:variable name="urlFoto">
 <xsl:value-of select="$host"/>
 </xsl:variable>
 <tr>
 <td>
 <a>
 <xsl:attribute name="href">
 <xsl:value-of select="$urlFoto"/>
 </xsl:attribute>
 <xsl:value-of select="."/>

 </td>
 <td>
 <xsl:value-of select="$urlFoto"/>
 </td>
 </tr>
</xsl:template>
```

http://www.mijnfoto.be  
http://www.mijnfoto.be/  
IMG003

attribuut toevoegen  
aan huidige element

```
<foto bron="IMG001">
 Mijn huis
</foto>
<foto bron="IMG002">
 Kinderen
</foto>
```

## Waarde select- en match-attribuut?

- Uitdrukkingen in XPath
- Voorbeelden
  - /
  - .
  - count(books/book)
  - title
  - servers/server

Industrieel Ingenieur Informatica, UGent

27

## XPath

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

Industrieel Ingenieur Informatica, UGent

28

## XPath

- Taal om stukken van een XML-document te identificeren
- XSLT 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

Industrieel Ingenieur Informatica, UGent

29

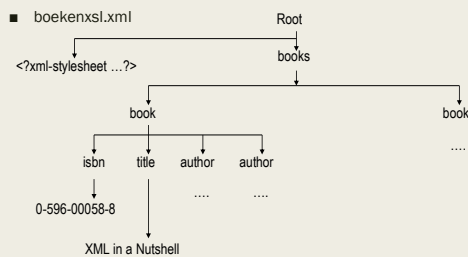
## XPath: Knopen

- Mogelijke knopen
  - Basisknoop (root node)
  - Elementen
  - Tekst
  - Attributen
  - Commentaar
  - Processing Instructions
  - Namespace
- Geen knopen
  - CDATA
  - Entiteiten
  - DTD

Industrieel Ingenieur Informatica, UGent

30

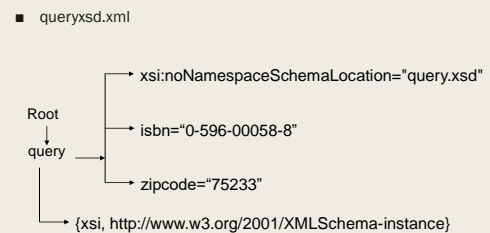
## Voorbeeld XPath-boom



Industrieel Ingenieur Informatica, UGent

31

## Voorbeeld XPath-boom



Industrieel Ingenieur Informatica, UGent

32

## XPath

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

Industrieel Ingenieur Informatica, UGent

33

## 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

Industrieel Ingenieur Informatica, UGent

34

## XPath: Padstructuur

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

Industrieel Ingenieur Informatica, UGent

35

## 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

```

<xsl:template match="author">
...
 <xsl:apply-templates select="name" />
</xsl:template>

```

Industrieel Ingenieur Informatica, UGent

36

## XPath: mogelijke paden



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

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

Industrieel Ingenieur Informatica, UGent

37

## 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

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

Industrieel Ingenieur Informatica, UGent

38

## 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

Industrieel Ingenieur Informatica, UGent

39

## XPath: mogelijke paden - jokertekens



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

Industrieel Ingenieur Informatica, UGent

40

## XPath: mogelijke paden – meerdere mogelijkheden



- Meerdere mogelijkheden
  - |
  - Voorbeelden
    - isbn|title
    - /books/book/isbn|title

Industrieel Ingenieur Informatica, UGent

41

## 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

Industrieel Ingenieur Informatica, UGent

42

## XPath: Padstructuur

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

Industrieel Ingenieur Informatica, UGent

43

## XPath: paden met voorwaarden

- Voorwaarde
  - Verwijderd knopen uit de selectie die niet aan de voorwaarde voldoen
  - Tussen vierkante haken []
  - Logische uitdrukking
  - Operatoren
    - <, >, >=, <=, !=, =, or, and
    - Let op soms &lt;
  - Opmerking
    - Waarde van een element is de tekstinhoud van het element
    - Waarde van een attribuut is de waarde van het attribuut

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

Industrieel Ingenieur Informatica, UGent

44

## XPath: Padstructuur

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

Industrieel Ingenieur Informatica, UGent

45

## XPath: lange padnamen

- Tot nu toe
  - Afgekorte padnamen
  - Handig 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 ::

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

Industrieel Ingenieur Informatica, UGent

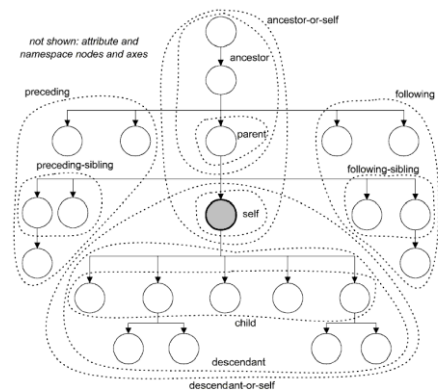
46

## XPath: lange padnamen - assen

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

Industrieel Ingenieur Informatica, UGent

47

bron: <http://krum.rz.uni-mannheim.de/inet-2004/sess-302.html>



## XPath



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

Industrieel Ingenieur Informatica, UGent

49

## 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

Industrieel Ingenieur Informatica, UGent

50

## XPath: uitdrukkingen



- Datatypes
  - *Getallen*
  - *Strings*
  - *Logische variabelen*
    - `true()`
    - `false()`
- Functies
  - *Knopen*
  - *Strings*
  - *Logische*
  - *Numerieke*

Industrieel Ingenieur Informatica, UGent

51

## XPath: Functies



- Voorbeelden
  - `position()`
  - `last()`: positie van de laatste knoop = aantal knopen in huidige knoop
  - `count(arg)`: aantal knopen bepaald door het argument *arg*
  - `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 functies ([https://www.w3schools.com/xml/xsl\\_functions.asp](https://www.w3schools.com/xml/xsl_functions.asp))

Industrieel Ingenieur Informatica, UGent

52

## Doel



- Basisconcepten XSLT kennen
  - Wat is XSLT?
  - Hoe werkt het?
- Structuur XSLT-document kennen en toepassen
- Basisprincipes XPath kennen
  - Wat is XPath?
  - Rol XPath in XSLT
  - Structuur van een pad
  - Verschillende assen

Industrieel Ingenieur Informatica, UGent

53