DMT Seminar

XSLT 2

Fleur Praal – 1 November 2016

XSLT

XSL = eXstensible Stylesheet Language, XML
XSLT = XSL Transformations

Transforms XML into X(HT)ML, PDF, MARC, etc. by working its way through the structure of the encoding

→ Build result-tree from source-tree; with XSL-file that is a tree itself!

XSLT processors
open source: XALAN, SAXON (built-in in Oxygen)

Getting started with XSLT

1. Study the XML source document structure

2. Outline the intended XML result document structure
   - translation step: from content to element names

3. Create XSL-file
   - specify it is a stylesheet *(let Oxygen help you)*
   - create template to match the root of source document:

4. Fill XSL-file

5. Configure transformation scenario

6. Check result

7. Repeat steps 4-6, if necessary ☺

Step 3: declare stylesheet and open template

`<xsl:stylesheet>`

- XSLT root element, opens stylesheet

`<xsl:template>` `@match`

- takes source-XML's root as value, to specify path for transformation

Step 4: Build XSL-file

<xsl:value-of> @select

- to select values of elements from source document and inserts into result tree;
- note that the paths in the select-attribute must depart from the element mentioned in the match-attribute of the template

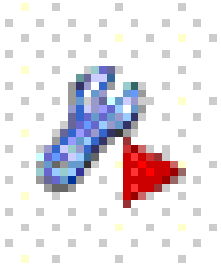
```
<xsl:value-of select="title"/>
```

<xsl:text>

- to insert literal text into the result tree

```
<xsl:text>This sentence will be visible in the result.</xsl:text>
```

Step 5: configure transformation scenario



- ‘Configure transformation scenario’
  - create new scenario
    - on tab ‘XSLT’ specify input XML and transformation XSL
    - on tab ‘Output’, configure output
  - save scenario

- execute scenario: output file will be created as specified

See also: XSLT Course – Appendix A, ‘Transformations in Oxygen’
http://www.bookandbyte.org/DMT/XSLT/index.php?file=app1.html

XSLT stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

    <xsl:template match="letter">

      <html>
                <head>
                        <title>
                                <xsl:text>XSLT transformation</xsl:text>
                        </title>
                </head>
                <body>
                        <h2>
                                <xsl:value-of select="head">
                        </h2>
                </body>
       </html>
    </xsl:template>

</xsl:stylesheet>
```

Exercise 1

Exercise 1

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

    <xsl:template match="letter">
        <html>
            <head/>
            <body>
                <h2><xsl:value-of select="head/title"/>
                </h2>
                <p><xsl:value-of select="body/greeting"/>
                </p>
            […]
            </body>
        </html>
    </xsl:template>

</xsl:stylesheet>
```

X-Path

/ navigate down

// hierarchy irrelevant

. highest established element

* any direct child

../ jump one level up

@ navigate to attribute

[criterion]

Exercise 2

Exercise 2

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

    <xsl:template match="literatureList">

      <ul>
        <xsl:for-each select="item">
              <li>
                    <xsl:value-of select="fullTitle"/>
              /li>
        </xsl:for-each>
      </ul>
    </xsl:template>

</xsl:stylesheet>
```

Exercise 2

XSLT elements (4)

`<xsl:sort> @select @data-type @order`

- to sort a list alphabeticaly or numerically, by the attribute specified
- note that <xsl:sort> must be the direct child of <xsl:for-each> (or <xsl:apply-templates>)

```
<xsl:sort select="fullTitle"
data-type="text" order="ascending"/>
```

Exercise 3

Exercise 3

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">

    <xsl:template match="literatureList">

      <ul>
        <xsl:for-each select="item">
        <xsl:sort select="fullTitle"/>
                <li>
                        <xsl:value-of select="fullTitle"/>
                /li>
        </xsl:for-each>
      </ul>
    </xsl:template>

</xsl:stylesheet>
```

Exercise 4

X-Path

/ navigate down

// hierarchy irrelevant

. highest established element

* any direct child

../ jump one level up

@ navigate to attribute

[criterion]

XPath: please note

- Be precise in nesting quotation marks (single/double), brackets (round, square, angular)
  `(XPath function)`
  `[criterion]`
  `<element>`
- Escape entities: `>`
- Remain aware of context
- Combining criteria: `[criterion='x' and criterion='y']`