

Semantic Web

XHTML

F. Abel, D. Krause

IVS Semantic Web Group

01.11.2007

From G. Antoniou, Frank v. Harmelen: A Semantic Web Primer

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE library PUBLIC "library.dtd">
<library location="Bremen">

  <author name="Henry Wise">
    <book title="Artificial Intelligence"/>
    <book title="Modern Web Services"/>
    <book title="Theory of Computation"/>
  </author>
  <author name="William Smart">
    <book title="Artificial Intelligence"/>
  </author>
  <author name="Cynthia Singleton">
    <book title="The Semantic Web"/>
    <book title="Browser Technology Revised"/>
  </author>

</library>
```

Excercise 1: Addressing and Querying XML

- Address all author elements:
- Address all author elements, that are children of the library:
- Address the location attribute nodes within library element nodes:
- Address all title attribute nodes within book elements anywhere in the document, which have the value "Artificial Intelligence":
- Address all books with the title "Artificial Intelligence":
- Address the first author element node in the XML document:
- Address the last book element within the first author element node in the document:
- Address all book element nodes without a title attribute:

Exercise 1: Addressing and Querying XML

- Address all author elements: `//author`
- Address all author elements, that are children of the library:
`library/author`
- Address the location attribute nodes within library element nodes:
`/library/@location`
- Address all title attribute nodes within book elements anywhere in the document, which have the value "Artificial Intelligence":
`//book[@title[.="Artificial Intelligence"]`
- Address all books with the title "Artificial Intelligence":
`//book[@title="Artificial Intelligence"]`
- Address the first author element node in the XML document:
`//author[1].`
- Address the last book element within the first author element node in the document: `//author[1]/book[last()]`
- Address all book element nodes without a title attribute:
`//book[not(@title)]`

Excercise 2: XSLT

- Visualize the following XML document like Amazon does.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="style.xsl"?>
<books>
  <book>
    <name link="http://www.amazon.com/.../0789722429/">
      XML by Example
    </name>
    <author>Benoit Marchal</author>
    <listprice>24.99</listprice>
    <price>17.49</price>
    <review>4.5</review>
    <publish>QUE</publish>
  </book>
  ...
</books>
```

Exercise from: <http://www.topxml.com/xsl/tutorials/intro/>

Excercise 2: XSLT

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

<xsl:template match="/">
  <html><head><title>Bookshop</title></head>
  <body>
    <xsl:apply-templates select="books"/>
  </body>
</html>
</xsl:template>

<xsl:template match="book">
  <a href="{name/@link}"><xsl:value-of select="name"/></a>
  <xsl:apply-templates select="author">
    <xsl:sort data-type="string" select="."/>
  </xsl:apply-templates>
  <span style="color:red"><xsl:value-of select="price"/></span>
  (<xsl:value-of select="listprice"/>)
  <br/>
</xsl:template>

</xsl:stylesheet>
```

Excercise 3: XSLT

Look at the course XML document:

<http://www.kbs.uni-hannover.de/Lehre/semweb07/inhalt.xml>

- Restyle the course homepage: Reorder the lessons according to their title.
- Number the lessons with 1., 2., 3. and so on
- Create a hyperlink which contains the text of the title and links to the GroupMe! group
- Display in a new line the keywords of the lesson.