



4 : Fundamentals of XML

IT2406 - Web Application Development 1

Level I - Semester 2

XML Stylesheets

What Is It?

- Extensible Stylesheet Language (XSL)
- Language for document transformation
 - Transformation (XSLT)
 - Converting XML to another form
 - **e.g. Use Xalan to convert to text, XML, HTML, etc...**
 - Formatting objects (XSL:FO)
 - Layout of XML document
 - **e.g. Use FOP to convert to PDF, etc...**
- Defined by W3C

The Simplest Stylesheet (1 of 2)

- Empty XSLT document

```
01    <?xml version='1.0' encoding='UTF-16'?>
02    <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
03                version='1.0'>
04    </xsl:stylesheet>
```

The Simplest Stylesheet (2 of 2)

- Empty XSLT document

```
01      <?xml version='1.0' encoding='UTF-16'?>
02      <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'
03                      version='1.0'>
04      </xsl:stylesheet>
```

- *Note:* This will simply copy the text content of the input document to the output.

What Are Its Features?

- Features
 - Templates
 - Map input patterns to output
 - Conditionals
 - Loops
 - Functions
 - Extensions

Templates

- Transformation rules
 - Selects nodes from input document
 - Uses XPath expressions
 - Specifies output
 - Can be text, XML, HTML (xml with html tags)
- Template modes
 - Allows multiple templates to have same rules
- Named templates
 - Call templates by name

Simple Template (1 of 4)

- Match document root
 - `<xsl:template match="/">`
 - `<HTML>`
 - `<BODY>`
 - **`<xsl:apply-templates select='child::*'/>`**
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`

Simple Template (2 of 4)

- Match document root
 - `<xsl:template match='/'>`
 - `<HTML>`
 - `<BODY>`
 - **`<xsl:apply-templates select='child::*'/>`**
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`

Simple Template (3 of 4)

- Match document root
 - `<xsl:template match="/">`
 - `<HTML>`
 - `<BODY>`
 - `<xsl:apply-templates select='child::*'/>`
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`

Simple Template (4 of 4)

- Match document root
 - `<xsl:template match="/">`
 - `<HTML>`
 - `<BODY>`
 - `<xsl:apply-templates select='child::*'/>`
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`

Simple Template+ (1 of 5)

- Match document root
 - `<xsl:template match='/'>`
 - `<HTML>`
 - `<BODY>`
 - **`<xsl:apply-templates select='child::*'/>`**
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`
 - `<xsl:template match='book'>`
 - `<H1> <xsl:value-of select='title'/> </H1>`
 - `</xsl:template>`

Simple Template+ (2 of 5)

- Match document root
 - `<xsl:template match='/'>`
 - `<HTML>`
 - `<BODY>`
 - **`<xsl:apply-templates select='child::*'/>`**
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`
 - `<xsl:template match='book'>`
 - `<H1> <xsl:value-of select='title'/> </H1>`
 - `</xsl:template>`

Simple Template+ (3 of 5)

- Match document root
 - `<xsl:template match='/'>`
 - `<HTML>`
 - `<BODY>`
 - **`<xsl:apply-templates select='child::*'/>`**
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`
 - `<xsl:template match='book'>`
 - `<H1> <xsl:value-of select='title'/> </H1>`
 - `</xsl:template>`

Simple Template+ (4 of 5)

- Match document root
 - `<xsl:template match='/'>`
 - `<HTML>`
 - `<BODY>`
 - **`<xsl:apply-templates select='child::*'/>`**
 - `</BODY>`
 - `</HTML>`
 - `</xsl:template>`
 - `<xsl:template match='book'>`
 - `<H1> <xsl:value-of select='title'/> </H1>`
 - `</xsl:template>`

Simple Template+ (5 of 5)

- Input

```
01 • Output <book isdn='1234-5678-90'>
02     <title>Care and Feeding of Wombats</title>
03     <chapter/>
nn    </book>
```

```
01     <HTML>
02     <BODY>
03         <H1>Care and Feeding of Wombats</H1>
04     </BODY>
05 </HTML>
```


Conditionals

- If statement
 - `<xsl:if test=' expression'> ... </xsl:if>`
- Switch statement
 - `<xsl:choose>`
 - `<xsl:when test=' expression'> ... </xsl:when>`
 - `<xsl:otherwise> ... </xsl:otherwise>`
 - `</xsl:choose>`
- *Predicates*
 - `foo[@bar="value"]`

Loops

- For statement
 - `<xsl:for-each select=' expression
 - <xsl:sort select=' expression
 - ...`
 - `</xsl:for-each >`

XPath Functions

- Node-set functions
 - e.g. `position()`, `last()`, `local-name()`, etc...
- String functions
 - e.g. `string()`, `contains()`, `substring()`, etc...
- Boolean functions
 - e.g. `boolean()`, `not()`, etc...
- Number functions
 - e.g. `number()`, `sum()`, `round()`, etc...

XSLT Functions

- Multiple source documents
 - e.g. `document("document2.xml")`
- Keys
- Number formatting
- Additional functions
 - e.g. `current()`, `system-property()`, etc...

Example Transformation (0 of 14)

- Source

```
01 <order>
02   <item code='BK123'>
03     <name>Care and Feeding of Wombats</name>
04     <price currency='USD'>42.00</price>
05   </item>
06 </order>
```

- Destination

```
01 <HTML>
02   <BODY>
03     <TABLE border='1'>
04       <TR> <TH>Item</TH> <TH>Price</TH> </TR>
05       <TR>
06         <TD>BK123 - <U>Care and Feeding of Wombats</U></TD>
07         <TD>$42.00</TD>
08       </TR>
09     </TABLE>
10   </BODY>
11 </HTML>
```

Example Transformation (1 of 14)

- Match <order> element

```
01 <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'  
02           version='1.0'  
03  
04 <xsl:template match='/order'>  
05   <HTML>  
06     <BODY>  
07       <TABLE border='1'>  
08         <TR> <TH>Item</TH> <TH>Price</TH> </TR>  
09         <xsl:for-each select='item'>  
10           <xsl:apply-templates/>  
11         </xsl:for-each>  
12       </TABLE>  
13     </BODY>  
14   </HTML>  
15 </xsl:template>
```

Example Transformation (2 of 14)

- Match <order> element

```
01 <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'  
02           version='1.0'>  
03  
04   <xsl:template match='/order'>  
05     <HTML>  
06       <BODY>  
07         <TABLE border='1'>  
08           <TR> <TH>Item</TH> <TH>Price</TH> </TR>  
09           <xsl:for-each select='item'>  
10             <xsl:apply-templates/>  
11           </xsl:for-each>  
12         </TABLE>  
13       </BODY>  
14     </HTML>  
15   </xsl:template>
```

Example Transformation (3 of 14)

- Match <order> element

```
01 <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'  
02           version='1.0'  
03  
04 <xsl:template match='/order'>  
05   <HTML>  
06     <BODY>  
07       <TABLE border='1'>  
08         <TR> <TH>Item</TH> <TH>Price</TH> </TR>  
09         <xsl:for-each select='item'>  
10           <xsl:apply-templates/>  
11         </xsl:for-each>  
12       </TABLE>  
13     </BODY>  
14   </HTML>  
15 </xsl:template>
```


Example Transformation (4 of 14)

- Match <order> element

```
01 <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'  
02           version='1.0'  
03  
04 <xsl:template match='/order'>  
05   <HTML>  
06     <BODY>  
07       <TABLE border='1'>  
08         <TR> <TH>Item</TH> <TH>Price</TH> </TR>  
09         <xsl:for-each select='item'>  
10           <xsl:apply-templates/>  
11         </xsl:for-each>  
12       </TABLE>  
13     </BODY>  
14   </HTML>  
15 </xsl:template>
```

Example Transformation (5 of 14)

- Match <order> element

```
01 <xsl:stylesheet xmlns:xsl='http://www.w3.org/1999/XSL/Transform'  
02           version='1.0'  
03  
04 <xsl:template match='/order'>  
05   <HTML>  
06     <BODY>  
07       <TABLE border='1'>  
08         <TR> <TH>Item</TH> <TH>Price</TH> </TR>  
09         <xsl:for-each select='item'>  
10           <xsl:apply-templates/>  
11         </xsl:for-each>  
12       </TABLE>  
13     </BODY>  
14   </HTML>  
15 </xsl:template>
```

Example Transformation (6 of 14)

- Match <item> element

```
17 <xsl:template match='item'>
18   <TR>
19     <TD>
20       <xsl:value-of select='@code'/'> -
21       <U> <xsl:value-of select='name'/'> </U>
22     </TD>
23     <xsl:apply-templates select='price'/'>
24   </TR>
25 </xsl:template>
```

Example Transformation (7 of 14)

- Match <item> element

```
17 <xsl:template match='item'>
18   <TR>
19     <TD>
20       <xsl:value-of select='@code'/'> -
21       <U> <xsl:value-of select='name'/'> </U>
22     </TD>
23     <xsl:apply-templates select='price'/'>
24   </TR>
25 </xsl:template>
```

Example Transformation (8 of 14)

- Match <item> element

```
17 <xsl:template match='item'>
18   <TR>
19     <TD>
20       <xsl:value-of select='@code'/'> -
21       <U> <xsl:value-of select='name'/'> </U>
22     </TD>
23     <xsl:apply-templates select='price'/'>
24   </TR>
25 </xsl:template>
```

Example Transformation (9 of 14)

- Match <item> element

```
17 <xsl:template match='item'>
18   <TR>
19     <TD>
20       <xsl:value-of select='@code'/'> -
21       <U> <xsl:value-of select='name'/'> </U>
22     </TD>
23     <xsl:apply-templates select='price'/'>
24   </TR>
25 </xsl:template>
```

Example Transformation (10 of 14)

- Match <price> element

```
27 <xsl:template match='price'>
28   <TD>
29     <xsl:choose>
30       <xsl:when test='text()="JPN"'> ¥ </xsl:when>
31       <xsl:otherwise> $ </xsl:otherwise>
32     </xsl:choose>
33     <xsl:value-of select='text()'/>
34   </TD>
35 </xsl:template>
36
37 </xsl:stylesheet>
```

Example Transformation (11 of 14)

- Match <price> element

```
27 <xsl:template match='price'>
28   <TD>
29     <xsl:choose>
30       <xsl:when test='text()="JPN"'> ¥ </xsl:when>
31       <xsl:otherwise> $ </xsl:otherwise>
32     </xsl:choose>
33     <xsl:value-of select='text()'/>
34   </TD>
35 </xsl:template>
36
37 </xsl:stylesheet>
```


Example Transformation (12 of 14)

- Match <price> element

```
27 <xsl:template match='price'>
28   <TD>
29     <xsl:choose>
30       <xsl:when test='text()="JPN"'> ¥ </xsl:when>
31       <xsl:otherwise> $ </xsl:otherwise>
32     </xsl:choose>
33     <xsl:value-of select='text()'/>
34   </TD>
35 </xsl:template>
36
37 </xsl:stylesheet>
```

Example Transformation (13 of 14)

- Match <price> element

```
27 <xsl:template match='price'>
28   <TD>
29     <xsl:choose>
30       <xsl:when test='text()="JPN"'> ¥ </xsl:when>
31       <xsl:otherwise> $ </xsl:otherwise>
32     </xsl:choose>
33     <xsl:value-of select='text()'/>
34   </TD>
35 </xsl:template>
36
37 </xsl:stylesheet>
```

Example Transformation (14 of 14)

- Output

Item	Price
BK123 - <u>Care and Feeding of Wombats</u>	\$42.00

Rendering XML in Browsers

- Latest browsers (e.g. IE 6.0+) have support for XSLT
- Insert "xml-stylesheet" processing instruction
 - `<?xml-stylesheet type='text/xsl' href='style.xsl'?>`
- Output

Item	Price
BK123 - <u>Care and Feeding of Wombats</u>	\$42.00

Useful Links

- XPath 1.0 Specification
 - <http://www.w3.org/TR/xpath>
- XSLT 1.0 Specification
 - <http://www.w3.org/TR/xslt>