UNIT - 5 XSL (Extensible StyleSheet Language)

- Need of XSL
- XSL: Three Parts
- XSLT Language Characteristics
- XSLT Features
- XSL Transformation (XSLT)
- XSM To HTML Transformation
- Looping
- Conditional Processing
- Numbers and Sorting

Need of XSL

XML Do

- Capturing and storing data
- Providing data validation facilities
- Make data handling platform independent
- Make data exchange easier

XSL Do

- Transforming data from one format to another
- Selecting and reorganising parts of an XML document in adesired manner
- Adding new contents to an XML document
- Formating XML documents as per the needs



Need of XSL

- XSL is a language.
- It uses stylesheets for manipulating XML documents.
- It can be said that **XSL** is made up of two different but related technologies.
 - XSLT (XSL Transformation)
 - XSL-FO (XSL Formatting Object)



Need of XSL

- XSLT: allows formating and transformation of XML documents. Eg: from one format to another
- XSL Formating objects(XSL-FO): it is similar to CSS. Uses for display of XML document contents.



XSL: Three Parts

• XSL Provides the action to XML . There are several technologies that fall under the XSL umbrella

- XSLT

A language to transform XML documents

- XSL Formatting Object

• A vocabulary for formatting specifics which is part of the actual XSL specification

- Xpath

A language for addressing any part of an XML document.



XSL Characteristics

- XSLT uses XML syntax
- No side-effects
 - Parts of stylesheet can be processed in any order and independently, cannot update the value of variable
- Pattern-matching rule based/declarative
 - Stylesheet is a templete of rules
 - No ordering is needed
 - Declaration is needed
 - Depends on X-path for large part



XSL Characteristics

Closure

- Output has same data structure as input
- Tree representation
- Output of 1 operation->input of another

Recursive

- With build in constructs



XSL Features

- Multiple input sources
- Ability to select document fragments using Xpath expressions
- Named and/or pattern based templates
- Parameterized templates
- Intermediate transformation state may be managed using variables
- Stylesheets may be combuned using include or import





XSL Features

Built in support for output sorting and numbering

- Both XML and non XML output is supported
- XSL processor extensions are supported withput side effects on core function



XSLT

- As the name suggest, XSLT is used to transform XML data from one structure to another.
- The transformation can be with respect to the following:
 - Create new content
 - Add to the exsting content
 - Modify XML content into another XML format
 - Perform operations such as searching
 - Extract parts of a document
 - Transform data into suitable output format
 - Convert text data into voice XML



XSLT

• XSLT Flow

```
Input XML----> XSLT ----> output document processor XML document
```

XSLT Stylesheet



XSLT

Transformating XML into HTML



XSL Transformation - Template

- XSLT is used to transform one XML document from one from to another.
- The result of applying XSLT to an XML document could be another XML, HTML, text and another documents.
- Templates
 - An XSLT document is an XSLT document which has:
 - A root element called Stylesheet
 - A file extension of .xsl



Creating Elements & Attributes

```
• <?xml version="1.0"?>
  <students>
   <student f_name="Raju">
     <id> 101 </id>
     <remarks> ..... </remarks>
  </student>
  <student f_name="Rakesh">
     <id> 101 </id>
     <remarks> ..... </remarks>
  </student>
```



XSLT Document

```
• <xsl:stylesheet version="1.0" xmlns:xsl="</p>
  http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <xsl:apply-templates/>
  </xsl:template>
  <xsl:template match="students">
    <students>
    <xsl:apply-templates/>
    </students>
  </xsl:template>
```



XSLT Document

```
<xsl:template match="student">
  <xsl:element name="{@f_name}">
  <xsl:attribute name="id"><xsl:value-of select="id"/></xsl:attribute>
<notes>
  <xsl:value-of select="remarks"/>
</notes>
</xsl:element>
</xsl:template>
</xsl:stylesheet>
```



<Raju id="101"><notes> sincere student</notes></Raju>



Looping

- Iterating through a list of items by using the <xsl:for-each> syntax
- <xsl:sort> for sorting detials



Conditional Processing

- Writing conditions using <xsl:if>
- <xsl:if test="condition">
- Writing multiple conditions using <xsl:when>
 - <xsl:choose>
 - <xsl:when test=condition>.....</xsl:when>



• Number:

<xsl:number level="multiple/single"
format"1/A/a">

• Variable:

- Defining variable
 - <xsl:variable name="name of var" select="..."/>
 - Using Variable
 - <xsl:value-of select="\$name of var" />



UNIT 5 COMPLETED

