XML-XSL-XLT

<xsl:output>

- The <xsl:output> element defines the format of the output document.
- Note: <xsl:output> is a top-level element, and must appear as a child node of <xsl:stylesheet> or <xsl:transform>.
- Syntax:-
- ```
  <xsl:output method="xml|html|text|name" version="string"
  encoding="string" omit-xml-declaration="yes|no"
  standalone="yes|no" doctype-public="string"
  doctype-system="string" cdata-section-elements="namelist"
  indent="yes|no" media-type="string"/>
  ```
  ```

Example

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>

...

...

</xsl:stylesheet>
```

Example

```
<?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" version="4.0" encoding="UTF-8"
indent="yes"/>

...

...

</xsl:stylesheet>
```

<xsl:text>

- The <xsl:text> element is used to write literal text to the output.
- This element may contain literal text, entity references, and #PCDATA.

```
<xsl:if test="position() < last()-1">
    <xsl:text>, </xsl:text>
  </xsl:if>
  <xsl:if test="position()=last()-1">
    <xsl:text>, and </xsl:text>
  </xsl:if>
  <xsl:if test="position()=last()">
    <xsl:text>!</xsl:text>
  </xsl:if>
```

Built-in Template Rules

- The first such built-in template rule is for element nodes and the root node. It works like this:

```
<xsl:template match="/|*">
  <xsl:apply-templates/>
</xsl:template>
```

- The second built-in template rule is for text and attribute nodes. It works as follows:

  <xsl:template match="text()">

    <xsl:value-of select="."/>

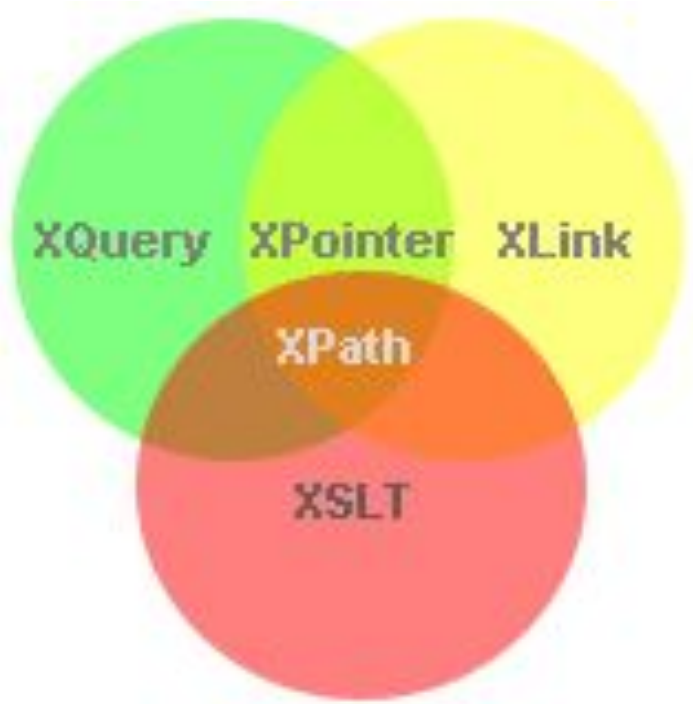
  </xsl:template>

- The built-in template for processing instructions and comments does nothing, as shown in the following:

  <xsl:template match="processing-instruction()|comment()"/>

Xpath

- XPath is a syntax for defining parts of an XML document
- XPath uses path expressions to navigate in XML documents
- XPath contains a library of standard functions
- XPath is a major element in XSLT and in XQuery
- XPath is a W3C recommendation



Symbol

Symbol	Description
//	Selects nodes in the document from the current node that match the selection no matter where they are
/	Selects the root node
tagname	Tag name of the current node
@	Select the attribute
attribute	Attribute name of the node
value	Value of the attribute

Types of XPath:

- **Absolute XPath:** Absolute XPath uses the root element of the HTML/XML code and followed by all the elements which are necessary to reach the desired element. It starts with the forward slash ‘/’

- **Relative XPath;** In this, XPath begins with the double forward slash ‘//’ which means it can search the element anywhere in the Webpage. Generally Relative Xpath is preferred as they are not complete path from Root node.

Commonly Used XPath Functions:

- **contains():** This Function is used to select the node whose specified attribute value contains the specified string provided in the function argument.
  - //input[contains(@id, 'fakebox')]
- **Starts-with():** This function is used to select the node whose specified attribute value starts with the specified string value provided in the function arguments.
  - //input[starts-with(@id, 'fakebox')]
- **text():** This function is used to find the node having the exact match with the specified string value in the function
  - //div

AND and OR in XPath

- AND and OR are used to combine two or more conditions to find the node.
- **Example:**
  - `//input[@value = 'Log In' or @type = 'submit']`

XPath Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element

XPath Expression	Result
//title[@lang]	Selects all the title elements that have an attribute named lang
//title[@lang='en']	Selects all the title elements that have a "lang" attribute with a value of "en"
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
/bookstore/book[price>35.00]/title	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00