

Overview

XSLT stands for **XSL Transformations**

XSLT uses XPath to navigate in XML documents

XSLT transforms an XML document into another XML document

To declare an XSL style sheet use one of the following

```
- <xs l:s tyl esheet versio n="1.0" xmlns: xsl ="ht tp: //w ww.w 3.o rg /19 99/ XSL /Tr ans for m">
- <xs l:t ran sform versio n="1.0" xmlns: xsl ="ht tp: //w ww.w 3.o rg /19 99/ XSL /Tr ans for m">
```

Note: `<xs l:s tyl esheet>` and `<xs l:t ran sfo rm>` are completely synonymous and either can be used!

XSL Elements

<code><xsl:stylesheet></code>	Defines this document as an XSLT style sheet
<code><xsl:template match="/"></code>	Used to build templates. Use the match attribute to select elements with XPath. Content defines HTML to output.
<code><xsl:value-of select="/"></code>	Used to extract the value of an XML element and add it to the output. Use the select attribute to select elements with XPath.
<code><xsl:for-each select="/"></code>	Used to select every XML element of a specified node-set. Wrap elements containing <code><xsl:value-of></code> (Loop)
<code><xsl:sort select="/"></code>	Used to sort the output.
<code><xsl:if test="expression"></code>	Used to put a conditional if test against the content of the XML file. Add inside for-each tags to add a condition to the loop.
<code><xsl:choose></code>	Used in conjunction with <code><xsl:when></code> and <code><xsl:otherwise></code> to express multiple conditional tests. Wrap around these tags like if {...} else {...}.
<code><xsl:when test="expression"></code>	The IF or ELSE IF component of a choose statement. Multiple can be used back to back within <code><xsl:choose></code>
<code><xsl:otherwise></code>	The ELSE component of a choose statement.
<code><xsl:apply-templates select="/"></code>	Used to apply a template to the current element or to the current element's child nodes.

Note: You can filter the output from the XML file by adding criterion to the select attribute in the `<xsl:for-each>` element.

Example: `<xsl:for-each select ="ca tal og/ cd[art ist ='Bob Dylan']">`

Legal operators are =, !=, < (<), > (>)

Template/Apply-Templates Example

```
<xsl:template match="/">
  <html>
  <body>
    <h2>My CD Collec tio n</ h2>
    <xsl:apply-templates/>
  </body>
</html>

</xsl: template>

<xsl:template match= " cd">
  <p>
    <xsl:apply-templates select ="ti tle " />
    <xsl:apply-templates select ="ar tis t"/>
  </p>

</xsl: template>

<xsl:template match= " tit le">
  Title: <span style= " col or: #ff 000 0">
    <xsl:value-of select ="/ ></ spa n>
  <br />

</xsl: template>

<xsl:template match= " art ist ">
  Artist: <span style= " col or: #00 ff0 0">
    <xsl:value-of select ="/ ></ spa n>
  <br />

</xsl: template>
```

