XSL = Extensible Stylesheet Language (a summary from W3Schools and McGrath (2007)

XSLT = XSL Transformations (into other formats, e.g. PDF and XHTML)

XSL is to XML as CSS is to HTML

XSL includes XSLT, XPath (the means for navigating XML data trees) and XSL-FO (Formatting Objects)

XSLT requires

[1] an *.xsl stylesheet with one or other synonymous declaration

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

or:

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

[2] an XML document to transform with a link to the *.xsl

```
<?xml-stylesheet type="text/xsl" href="cdcatalog.xsl"?>
```

XSL <template>

<xsl:template match="/"> where the value of the match attribute is the XPath expression.

<xsl:value-of> to extract data

e.g. <td><xsl:value-of select="catalog/cd/title"/></td> where the value of the select attribute is the XPath expression.

<xsl:for-each>

e.g. <xsl:for-each select="catalog/cd"> where the value of the select attribute is the XPath expression.

<xsl:sort >

e.g. <xsl:sort select="artist"/> where the value of the select attribute is the XPath expression.

<xsl: if> conditional test

e.g. <xsl:if test="price > 10">

<xsl:choose>

Syntax ...

```
<xsl:choose>
  <xsl:when test="expression">
    ... some output ...
  </xsl:when>
  <xsl:otherwise>
    ... some output ....
  </xsl:otherwise>
</xsl:choose>
```

<xsl:apply-templates>

Syntax example ...

```
<xsl:template match="cd">
  <p>
  <xsl:apply-templates select="title"/>
  <xsl:apply-templates select="artist"/>
  </p>
</xsl:template>

<xsl:template match="title">
  Title: <span style="color:#ff0000">
  <xsl:value-of select="."/></span>
  <br />
</xsl:template>
```