Xml xslt 트랜스폼**XSLTransformer**

**try** {

***Source*** xmlSource = **new** **StreamSource**("./xml/catalog.xml");

***Source*** xsltSource = **new** **StreamSource**("./xml/catalog.xsl");

***Result*** htmlResult = **new** **StreamResult**("./xml/catalog.html");

**TransformerFactory** tFactory = **TransformerFactory**.*newInstance*();

**Transformer** transformer = tFactory.**newTransformer**(xsltSource);

transformer.**transform**(xmlSource, htmlResult);

} **catch**(**Exception** e) {

}

Xml

<?xml version=*"1.0"* encoding=*"utf-8"* standalone=*"yes"*?>

<?xml-stylesheet type=*"text/xsl"* href=*"catalog.xsl"*?>

<Catalog>

<Book id=*"1861003323"*>

<Title part=*"XML"*>Professional VB6 XML</Title>

<Authors>

<Author>James Britt</Author>

<Author>Teun Duynstee</Author>

</Authors>

<Publisher>Wrox Press, Ltd.</Publisher>

<PubDate>March 2000</PubDate>

<ISBN>1-861003-32-3</ISBN>

<Price>49.99</Price>

<CoverImage>3323.gif</CoverImage>

</Book>

<Book id=*"1861003129"*>

<Title part=*"XSLT"*>XSLT Programmer's Reference</Title>

<Authors>

<Author>Michael Kay</Author>

</Authors>

<Publisher>Wrox Press, Ltd.</Publisher>

<PubDate>April 2000</PubDate>

<Pages>800</Pages>

<ISBN>1-861003-12-9</ISBN>

<Price>34.99</Price>

<CoverImage>3129.gif</CoverImage>

</Book>

</Catalog>

Xslt

<?xml version=*"1.0"*?>

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

version=*"1.0"*>

<xsl:output method=*"html"* />

<xsl:attribute-set name=*"imgAttrs"*>

<xsl:attribute name=*"width"*>150</xsl:attribute>

<xsl:attribute name=*"border"*>1</xsl:attribute>

</xsl:attribute-set>

<xsl:variable name=*"author"*>

<xsl:text>Author</xsl:text>

</xsl:variable>

<xsl:template match=*"/"*>

<HTML>

<HEAD>

<TITLE>Book Catalog</TITLE>

</HEAD>

<BODY>

<h2>Book Catalog</h2>

<hr color=*"black"* />

<xsl:apply-templates select=*"Catalog/Book"* />

</BODY>

</HTML>

</xsl:template>

<xsl:template match=*"Catalog/Book"*>

<xsl:variable name=*"titleNode"* select=*"Title"* />

<p>

<xsl:choose>

<xsl:when test=*"$titleNode/@part='XML'"*>

<font color=*"red"*>

<b><xsl:value-of select=*"$titleNode"* /></b>

</font>

</xsl:when>

<xsl:when test=*"$titleNode/@part='XSLT'"*>

<font color=*"blue"*>

<b><xsl:value-of select=*"$titleNode"* /></b>

</font>

</xsl:when>

<xsl:otherwise>

<font color=*"black"*>

<b><xsl:value-of select=*"$titleNode"* /></b>

</font>

</xsl:otherwise>

</xsl:choose>

</p>

<p>

<b><xsl:value-of select=*"$author"* /> : </b>

<xsl:for-each select=*"Authors/Author"*>

<xsl:sort select=*"."* order=*"descending"* />

<xsl:variable name=*"position"* select=*"position()"* />

<xsl:number value=*"$position"* format=*"1"*/>

<xsl:text>. </xsl:text>

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

<xsl:if test=*"$position != last()"*>, </xsl:if>

</xsl:for-each>

</p>

<xsl:call-template name=*"priceTemplate"* />

<p>

<img xsl:use-attribute-sets=*"imgAttrs"*>

<xsl:attribute name=*"src"*>

<xsl:value-of select=*"CoverImage"* />

</xsl:attribute>

</img>

</p>

</xsl:template>

<xsl:template name=*"priceTemplate"*>

<xsl:variable name=*"price"* select=*"Price"* />

<p>

<b>Price : </b>

<xsl:value-of select=*"concat('$ ', $price)"* />

</p>

</xsl:template>

</xsl:stylesheet>