

XSLT

eXtensible Stylesheet Language Transformations

w3c Schools

transforming each XML element into an (X)HTML
element.

XSLT is a language for transforming XML documents into other formats. Normally XSLT does this by transforming each XML element into an (X)HTML element.

XPath is a language for navigating in XML documents.

XQuery is a language for querying XML documents.

XSL on XAMP

XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

Modules

Service	Module	PID(s)	Port(s)	Actions
<input checked="" type="checkbox"/>	Apache	16604 1932	443, 8088	<input checked="" type="button" value="Stop"/> Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

오전 7:54:40 [main] XAMPP Installation Directory: "c:\xampp"

오전 7:54:40 [main] Checking for prerequisites

오전 7:54:41 [main] All prerequisites found

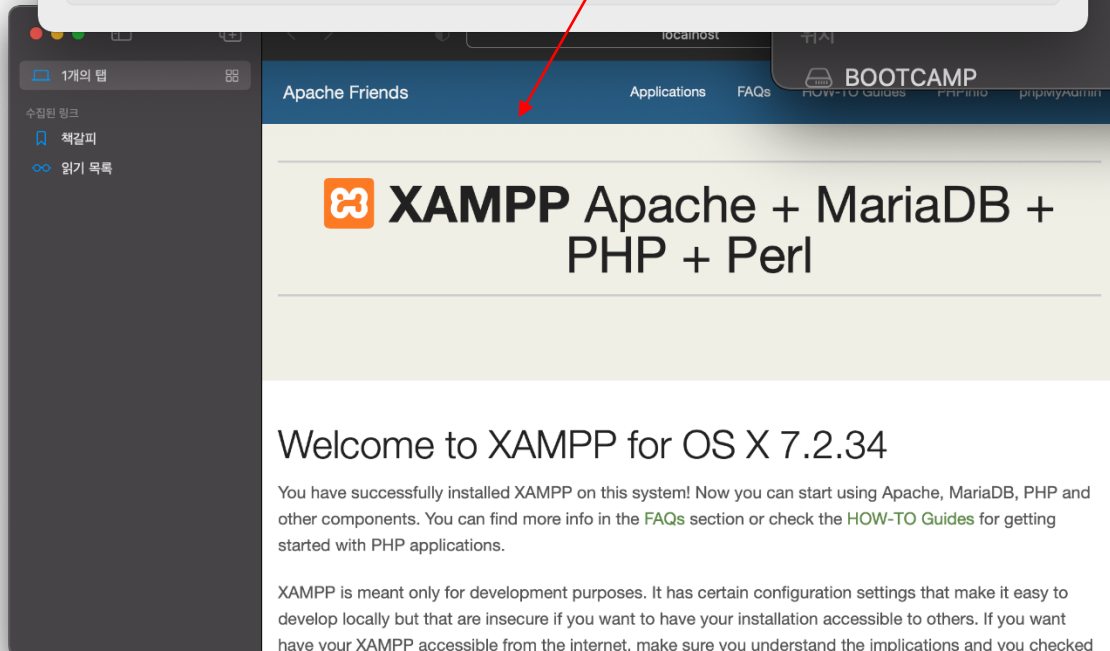
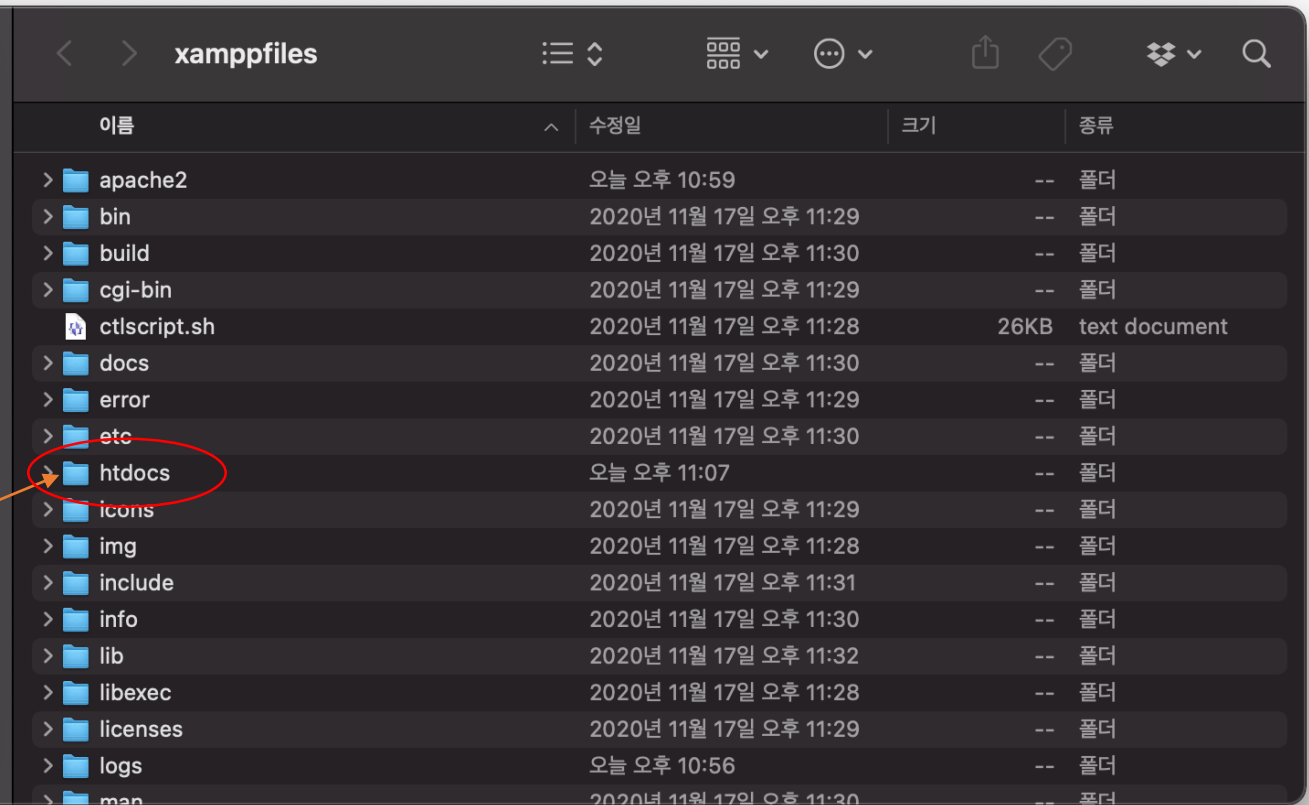
오전 7:54:41 [main] Initializing Modules

오전 7:54:41 [main] Starting Check-Timer

오전 7:54:41 [main] Control Panel Ready

오전 7:54:43 [Apache] Attempting to start Apache app...

오전 7:54:44 [Apache] Status change detected: running



XAMPP/htdocs/XSL

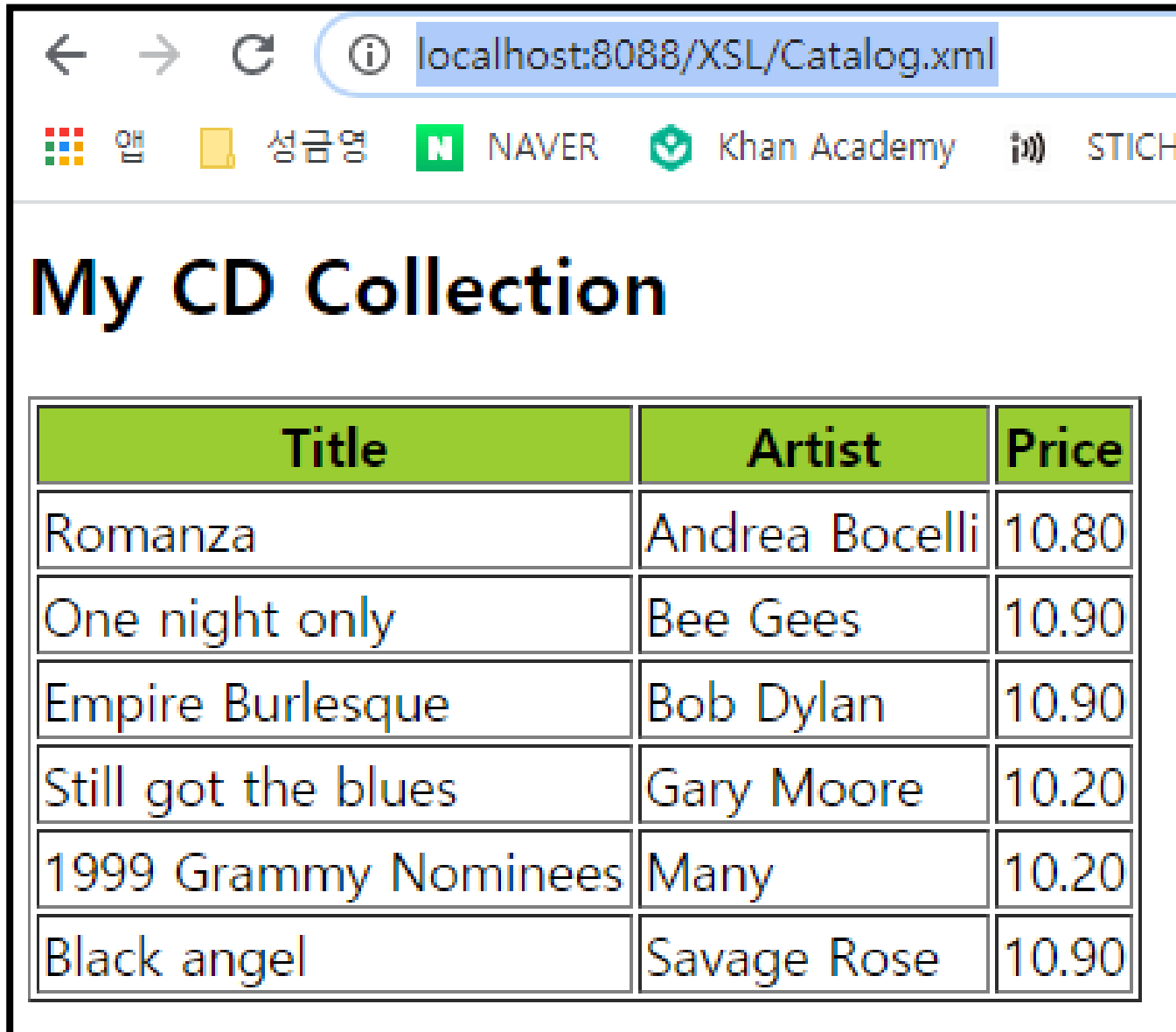
C:\xampp\htdocs\XSL

← → ↕ ↑ > > 내 PC > 로컬 디스크 (C:) > xampp > htdocs > XSL XSL 검색

이름	수정된 날짜	유형	크기
Catalog.xml	2019-08-13 오후 1:33	PHP 파일	
Catalog.xml	2019-10-12 오후 7:31	XML 문서	
Catalog.xml	2018-03-14 오후 1:01	XSL 스타일시트	
Catalog.xml	2019-10-15 오전 8:02	HTML 문서	
Catalog.xml	2019-10-10 오후 11:11	XMI 문서	

33개 항목 | 1개 항목 선택함 672바이트

http://localhost:8088/XSL/Catalog.xml



A screenshot of a web browser window. The address bar shows 'localhost:8088/XSL/Catalog.xml'. Below the address bar are several icons: a multi-colored square, a yellow folder icon, a green 'N' logo, a green checkmark icon, and a speaker icon. The main content area has the title 'My CD Collection' in a large, bold, black font. Below the title is a table with three columns: 'Title', 'Artist', and 'Price'. The table contains six rows of data.

Title	Artist	Price
Romanza	Andrea Bocelli	10.80
One night only	Bee Gees	10.90
Empire Burlesque	Bob Dylan	10.90
Still got the blues	Gary Moore	10.20
1999 Grammy Nominees	Many	10.20
Black angel	Savage Rose	10.90

Declaring an XSL style sheet...

```
<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">
```

Linking the XSL Style Sheet to the XML Document...

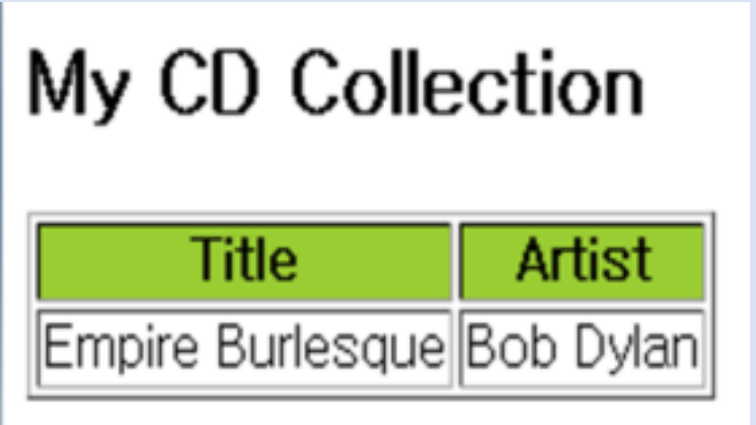
```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="Catalog.xsl"?>  
<catalog>  
  <cd>  
    <title>Empire Burlesque</title>  
    <artist>Bob Dylan</artist>  
    <country>USA</country>  
    ....
```


An XSL style sheet consists of one or more **set of rules** that are called **templates**.

- The **match attribute** is used to associate a template with an XML element.
- The value of the match attribute is an **XPath expression** (i.e. match="/" defines the **whole document**).

The `<xsl:value-of>` Element

```
<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
<table border="1">
  <tr bgcolor="#9acd32">
    <th>Title</th>
    <th>Artist</th>
  </tr>
  <tr>
    <td><xsl:value-of select="catalog/cd/title" /></td>
    <td><xsl:value-of select="catalog/cd/artist" /></td>
  </tr>
</table>
</body>
</html>
</xsl:template>
```



The screenshot shows the rendered output of the XSL-FO template. It features a heading "My CD Collection" followed by a table with two columns: "Title" and "Artist". The first row of data shows "Empire Burlesque" and "Bob Dylan".

My CD Collection	
Title	Artist
Empire Burlesque	Bob Dylan

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
    <body>
      <h2>My CD Collection</h2>
      <table border="1">
        <tr bgcolor="#9acd32">
          <th>Title</th>
          <th>Artist</th>
        </tr>
        <tr>
          <td> <xsl:value-of select="catalog/cd/title"/> </td>
          <td> <xsl:value-of select="catalog/cd/artist"/> </td>
        </tr>
      </table>
```







Catalog.xml

Catalog2.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type = "text/xsl" href = "Catalog2.xsl
```

Catalog.xml

Catalog2.xsl

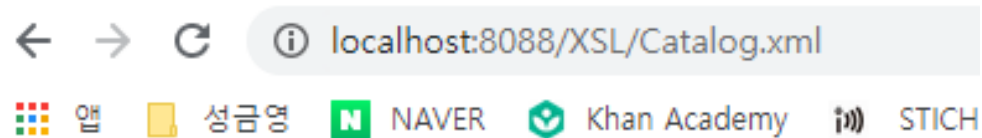
 localhost:8088/XSL/Catalog.xml	
 앱  성금영  NAVER  W3C SCHOOL 	
<h1>My CD Collection</h1>	
Title	Artist
Empire Burlesque	Bob Dylan

```
<xsl:for-each select="catalog/cd">
```

FOR Loop

```
<xsl:for-each select="catalog/cd">  
<tr>  
  <td><xsl:value-of select="title"/></td>  
  <td><xsl:value-of select="artist"/></td>  
</tr>
```

Catalog.xml



My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook
Maggie May	Rod Stewart
Romanza	Andrea Bocelli

```
<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th style="text-align:left">Title</th>
      <th style="text-align:left">Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Fix this!

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>.....
<table border="1">
  <tr bgcolor="#9acd32">
    <th>Title</th>
    <th>Artist</th>
  </tr>
  <xsl:for-each select="catalog/cd">
    <tr>
      <td><xsl:value-of select="catalog/cd/title"/></td>
      <td><xsl:value-of select="catalog/cd/artist"/></td>
    </tr>
  </xsl:for-each>
</table>
```

My CD Collection

[illegible]


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

```
<html>
```

```
<body>
```

```
<h2>My CD Collection</h2>
```

```
<table border="1">
```

```
<tr bgcolor="#9acd32">
```

```
<th style="text-align:left">Title</th>
```

```
<th style="text-align:left">Artist</th>
```

```
</tr>
```

```
<xsl:for-each select="catalog/cd">
```

```
<tr>
```

```
<td><xsl:value-of select="title " /> </td>
```

```
<td><xsl:value-of select="artist " /> </td>
```

```
</tr>
```

```
</xsl:for-each>
```

```
</table>
```

"for-each" element



localhost:8088/XSL/Catalog.xml



앱



성금영



NAVER



W3C SCHOOL



HISNet



My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore
Eros	Eros Ramazzotti
One night only	Bee Gees
Sylvias Mother	Dr.Hook

The <xsl:choose> Element

Syntax

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

```
<xsl:template match="/">
<html>
<body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="green">
      <th style="text-align:left">Title</th>
      <th style="text-align:left">Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <xsl:choose>
          <xsl:when test="price > 10">
            <td bgcolor="violet">
              <xsl:value-of select="artist"/></td>
          </xsl:when>
          <xsl:otherwise>
            <td><xsl:value-of select="artist"/></td>
          </xsl:otherwise>
        </xsl:choose>
      </tr>
    </xsl:for-each>
  </table>
```

My CD Collection

Title	Artist
Empire Burlesque	Bob Dylan
Hide your heart	Bonnie Tyler
Greatest Hits	Dolly Parton
Still got the blues	Gary Moore

```
<xsl:choose>
  <xsl:when test="price > 10">
    <td bgcolor="#ff00ff">
      <xsl:value-of select="artist"/></td>
    </xsl:when>
    <xsl:when test="price > 9">
      <td bgcolor="#cccccc">
        <xsl:value-of select="artist"/></td>
      </xsl:when>
      <xsl:otherwise>
        <td><xsl:value-of select="artist"/></td>
      </xsl:otherwise>
    </xsl:choose>
```