

XSLT 2.0 확장

(3주차)

학습개요

- 학습 목표

- XSLT 2.0에 대해 설명할 수 있다.
- Saxon-CE를 이용한 XSLT 2.0 문서 변환 프로세스를 설명할 수 있다.
- XSLT 2.0 스타일시트를 작성할 수 있다.

- 학습 내용

- XSLT 2.0과 Saxon-CE
- XSLT 2.0 스타일시트
- XSLT 2.0 스타일시트 실습

XSLT 2.0

- W3C는 2007년 1월 새로운 버전의 XSLT 2.0 발표
- XML 원본 문서의 콘텐츠 선택에 XPath 2.0과 XDM(XML Data Model)을 사용함
- www.w3.org/2007/01/qt-pressrelease
- <http://www.w3.org/TR/xslt20/>
- 웹 브라우저들은 XSLT 2.0을 아직 지원하지 않으므로 XSLT 2.0을 지원하는 XSLT 프로세서나 XML 전용에디터가 추가적으로 필요

XSLT 2.0 개발의 주요 목적

- XSLT 사용의 편의성을 개선
- 다중 출력 문서에 대한 지원
- 사용자 정의 함수 생성 및 호출에 대한 지원
- XML 스키마 사용에 대한 지원
- 콘텐츠 그룹핑을 단순화
- XSLT 1.0과의 하위 호환성 유지

XSLT 2.0 개발을 위한 준비

- oXygen XML editor, EditX XML Editor 등 다양한 상용 도구들을 이용해 XSLT 2.0 애플리케이션 개발이 가능
- Saxon 라이브러리를 통한 XSLT 2.0 프로세싱 지원
- 웹 브라우저들은 XSLT 2.0 미지원
- Saxon-CE(Client Edition)
 - 웹 브라우저에서 사용할 수 있는 XSLT 2.0 구현체
 - Saxon-CE 1.1부터 오픈소스 소프트웨어로 전환

Saxon-CE

SAXONICA
XSLT AND XQUERY PROCESSING

[Our Products](#) [Technology](#) [Download](#) [Purchase](#) [Documentation](#) [Support](#) [About us](#)

[SAXONICA > Our Products > Saxon-CE](#) [Our Products](#) | [Latest Releases](#) | [Why Saxon?](#) | [Feature Matrix](#) | [Saxon/C](#) | [Saxon-CE](#)

Saxon-CE

Saxon-CE (Client Edition) is Saxonica's implementation of XSLT 2.0 for use on web browsers.

The current release is Saxon-CE 1.1 (released 22 February 2013).

Unlike previous releases, Saxon-CE 1.1 is **open-source** software.

Product Information

Information about the general capabilities of the product, as well as detailed instructions for installing the product and writing applications, can be found in the [documentation](#) (for Javascript users, this is presented using a Saxon-CE browser application: it will take a few seconds to load the first time you use it).

A number of Saxon-CE users have built proof of concept [demonstrations](#) which use the Saxon-CE software.

[Return to top of page](#)

License

The product is open-source. The primary license is the Mozilla Public License version 2.0. In most cases, the source files are marked as being "incompatible with secondary licenses", which means that the code cannot be combined with code issued under a GPL license (the reason for this is the difficulty of contacting all past contributors to gain their assent).

On this page:

- [Product Information](#)
- [License](#)
- [Download](#)
- [Installation](#)
- [Sample Applications](#)
- [Support](#)
- [Source code](#)
- [Archive](#)
- [Related Products](#)

<http://www.saxonica.com/ce/index.xml>

Saxon-CE를 이용한 XSLT 2.0 개발

파일명: books.html

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>카탈로그</title>
  <link rel="stylesheet" href="books.css" type="text/css">
</head>
<body>
```

Saxon-CE 로딩

```
<script type="text/javascript" src="Saxonce/Saxonce.nocache.js"> </script>
```

```
<script type="application/xslt+xml" language="xslt2.0" src="books.xsl" data-source="books.xml"> </script>
```

```
<h1 id="title">카탈로그</h1>
```

XSLT 2.0 스크립트 엘리먼트 사용

```
<p>특정 장르의 책을 보려면 장르의 체크박스를 선택하십시오.</p>
```

```
<p>테이블 정렬을 하려면 컬럼의 헤더를 클릭하십시오.</p>
```

```
...
```

XSLT 2.0 스타일시트 적용과 결과 생성

파일명: books.xml

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:ixsl="http://saxonica.com/ns/interactiveXSLT"
  xmlns:prop="http://saxonica.com/ns/html-property"
  xmlns:style="http://saxonica.com/ns/html-style-property"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs prop"
  extension-element-prefixes="ixsl"
```

```
  version="2.0" >
```

XSLT 2.0 스타일시트에 대한 버전 기술

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

```
<xsl:result-document href="#title" method="ixsl:replace-content">
```

```
  오늘(<xsl:value-of select="format-date(current-date(), '[Y0001]/[M01]/[D01]')" />)의 책을 소개합니다
```

```
</xsl:result-document>
```

```
<xsl:result-document href="#books">
```

```
<table>
```

```
<thead>
```

...

오늘(2015/02/11)의 책을 소개합니다

특정 장르의 책을 보려면 장르의 체크박스를 선택하십시오.

데이터 정렬을 하려면 컬럼의 헤더를 클릭하십시오.

장르

코드	설명	
F	Fantasy	<input checked="" type="checkbox"/>
CR	Classical Romance	<input type="checkbox"/>
PT	Professional Textbook	<input checked="" type="checkbox"/>

책

제목	저자	출판사	출판일	ISBN	가
The Big Over Easy	Jasper Fforde	Hodder & Stoughton	2005-07-11	0340835672	7
The Eyre Affair	Jasper Fforde	Penguin	2003-02-25	0142001805	7
XSLT 2.0 Programmer's Reference	Michael Kay	Wrox	2004-10-13	0470192747	4
XSLT Cookbook	Sal Mangano	O'Reilly	2005-12-14	0596009747	3

결과 문서 생성
/컨텐츠 교체 및

XSLT 2.0 스타일시트

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="2.0">  
  
</xsl:stylesheet>
```

- XSLT 1.0 스타일시트와 기본 문서의 내용은 동일하며 version
 애트리뷰트의 값을 2.0으로 기술해야 함

단순화된 스타일시트

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

<html xmlns:xsl=http://www.w3.org/1999/XSL/Transform
      xsl:version="2.0">

  <head>
    <title>카탈로그</title>
  </head>

  <body>
    ...
  </html>
```

- <html> 태그를 바로 써서 <xsl:stylesheet>, <xsl:output>, <xsl:template> 엘리먼트를 제거할 수 있음
- 이 때 version 애트리뷰트에 반드시 XSLT 네임스페이스를 가리키는 xsl: 접두어를 붙여야 함

출력 문서 생성

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

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

```
<html>
```

```
<head> <title>학생 정보</title> </head>
```

```
...
```

```
</html>
```

```
</xsl:template>
```

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

```
<xsl:result-document method="html">
```

```
<html>
```

```
<head> <title>학생 정보</title> </head>
```

```
...
```

```
</html>
```

```
</xsl:result-document>
```

```
</xsl:template>
```

- <xsl:result-document> 엘리먼트를 이용해 결과 문서를 생성

다중 출력 문서

```
<xsl:result-document href="contacts.html" method="html">
...
</xsl:result-document>

<xsl:result-document href="#result" method="html">
...
</xsl:result-document>
```

- XSLT 1.0은 하나의 출력 문서만을 생성하지만, XSLT 2.0은 다중 출력 문서 생성을 지원함
- `<xsl:result-document>` 엘리먼트의 href 애트리뷰트의 값으로 출력할 URI를 지정
- 출력 문서가 필요한 만큼 `<xsl:result-document>` 엘리먼트를 사용할 수 있음

사용자 정의 함수

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
  version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:ijh="http://whatever">

  <xsl:function name="ijh:hello">
    안녕하세요.
  </xsl:function>

</xsl:stylesheet>
```

- 사용자 정의 함수를 위한 네임스페이스를 기술
- `<xsl:function>` 엘리먼트의 `name` 애트리뷰트의 값으로 사용자 정의 함수명을 기술
- 이 때 반드시 함수명에 네임스페이스 접두어를 기술해야 함
- 함수의 내용은 반환할 값을 XSLT 2.0과 XPath 2.0 명령으로 작성

사용자 정의 함수

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
  version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:ijh="http://whatever">
...
<xsl:function name="ijh:now">
  <xsl:variable name="now" select="adjust-dateTime-to-timezone(current-dateTime(),
    xs:dayTimeDuration('PT9H'))"/>
  현재 시각은 <xsl:value-of select="format-dateTime($now, '[Y0001]/[M01]/[D01]
    [H01]:[m01]:[s01]')"/> 입니다.
</xsl:function>

</xsl:stylesheet>
```

- 함수 정의시 변수가 필요할 경우 `<xsl:variable name="variable_name" select="expression">` 엘리먼트를 이용해 선언하고 `$variable_name`을 통해 그 값을 참조

사용자 정의 함수

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet
  version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:ijh="http://whatever">
...
  <xsl:function name="ijh:resize">
    <xsl:param name="size"/>
    <xsl:param name="times"/>
    <xsl:value-of select="$size * $times"/>
  </xsl:function>

</xsl:stylesheet>
```

- 함수 정의시 매개변수가 필요할 경우 `<xsl:param name="param_name">` 엘리먼트를 이용해 선언하고 `$param_name`을 통해 그 값을 참조

사용자 정의 함수

```
<xsl:template match="/">
  <xsl:result-document href="#udf1">
    <xsl:value-of select="ijh:hello()"/>
  </xsl:result-document>

  <xsl:result-document href="#udf2">
    <xsl:value-of select="ijh:now()"/>
  </xsl:result-document>

  <xsl:result-document href="#udf3">
    <xsl:apply-templates select="/udf/image">
    </xsl:apply-templates>
  </xsl:result-document>
</xsl:template>
```

```
<xsl:template match="image">
  <img>
    <xsl:attribute name="src">
      <xsl:value-of select="./@file"/>
    </xsl:attribute>
    <xsl:attribute name="width">
      <xsl:value-of select="ijh:resize(./@width, 3)"/>
    </xsl:attribute>
    <xsl:attribute name="height">
      <xsl:value-of select="ijh:resize(./@height, 3)"/>
    </xsl:attribute>
  </img>
</xsl:template>
```

- 사용자 정의 함수 호출 시 네임스페이스 접두어와 함께 함수명을 이용해 호출
- 입력 매개변수를 가진 함수는 인자 리스트를 전달

그룹핑

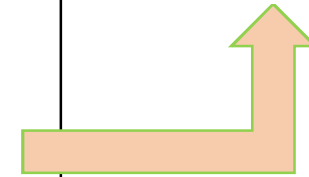
파일명: products.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<products>
  <product id="p1" name="텔레비전" price="2500000" stock="4" country="한국"/>
  <product id="p2" name="진공청소기" price="450000" stock="5" country="독일"/>
  <product id="p3" name="세탁기" price="850000" stock="19" country="독일"/>
  <product id="p4" name="컴퓨터" price="800000" stock="5" country="한국"/>
  <product id="p5" name="식탁" price="450000" stock="3" country="중국"/>
  <product id="p6" name="쇼파" price="1250000" stock="7" country="이탈리아"/>
  <product id="p7" name="책상" price="300000" stock="20" country="중국"/>
</products>
```

그룹핑

```
<xsl:result-document href="#result">
  <table border="1">
    <tr><th>제조국</th><th>제품명</th></tr>
    <xsl:for-each-group select="//product" group-by="@country">
      <xsl:sort select="@country" order="ascending" data-type="text"/>
      <tr>
        <td><xsl:value-of select="@country"/></td>
        <td><xsl:value-of select="current-group()/@name"/></td>
      </tr>
    </xsl:for-each-group>
  </table>
</xsl:result-document>
```

제조국	제품명
독일	진공청소기 세탁기
이탈리아	쇼파
중국	식탁 책상
한국	텔레비전 컴퓨터



- `<xsl:for-each-group select="expression" group-by="expression">` 엘리먼트의 select 애트리뷰트의 값으로 그룹핑할 노드들을 선택하고, 이 노드들에 대한 그룹핑 기준을 group-by 애트리뷰트의 값으로 지정
- `<xsl:for-each-group>` 엘리먼트 내에서 `current-group()`을 통해 현재 그룹을 반환

학습정리

- XSLT 2.0은 특정 정보를 찾기 위해 XPath 2.0과 XDM(XML Data Model)를 사용함
- Saxon-CE를 통해 웹 브라우저에서 XSLT 2.0 기능 사용
- XSLT 2.0은 다중 출력 문서를 지원함
- XSLT 2.0은 사용자 정의 함수를 지원함
- XSLT 2.0은 그룹핑이 쉬움