

# XSL-FO

(7주차)

# 학습개요

- 학습 목표

- XSL-FO 문서 구조를 설명할 수 있다.
- XSL-FO 프로세싱 과정을 설명할 수 있다
- XSL-FO 프로세서를 이용해 PDF문서를 생성하는 방법을 설명할 수 있다.

- 학습 내용

- XSL-FO 소개
- XSL-FO 프로세싱
- XSL-FO 문서
- XSLT와 XSL-FO
- XSL-FO 실습

# XSL-FO

- XSL Formatting Objects
- 페이지 레이아웃, 표지, 여백, 줄간격, 머리글, 바닥글, 미주, 각주 등 지원
- XSL-FO 문서를 작성해 인쇄용 문서(PDF 등)를 생성
- 인쇄 포맷터를 가진 XSL-FO 프로세서가 필요함
- <http://www.w3.org/TR/xsl/#fo-section>

# Apache™ FOP

- 대표적인 오픈소스 XSL-FO 프로세서
- Apache™ Software Foundation에 의해 관리되고 있는 XML Graphics Project의 하위 프로젝트
- 현재 FOP 1.1이 최신 버전임
- Java 애플리케이션 (JDK 1.6 이상 지원)
- <http://xmlgraphics.apache.org/fop/>



# Apache™ FOP와 PDF 문서 생성

```
<?xml version="1.0"?>
<root xmlns:fop="http://xml.apache.org/fop/1.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://xml.apache.org/fop/1.0 http://xml.apache.org/fop/1.0/fop.xsd">
  <fop:layout-master-set>
    <fop:layout for="first-page">
      <fop:sample-page-master>
        <fop:master-name>first</fop:master-name>
        <fop:region-before>margin</fop:region-before>
        <fop:region-after>exten</fop:region-after>
      </fop:sample-page-master>
    </fop:layout>
  </fop:layout-master-set>
  <fop:page-sequence-master>
    <fop:repeatable-page-master>
      <fop:condition>
        <fop:master-name>first</fop:master-name>
      </fop:condition>
    </fop:repeatable-page-master>
  </fop:page-sequence-master>
  <fop:page-sequence>
    <fop:page-master-name>first</fop:page-master-name>
  </fop:page-sequence>
</root>
```

Render



# XSL-FO 샘플 문서 - hello.fo

```
<?xml version="1.0" encoding="UTF-8"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="hello">
      <fo:region-body/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="hello">
    <fo:flow flow-name="xsl-region-body">
      <fo:block>
        Hello World!!!
      </fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
```

# Apache™ FOP 환경설정 및 실행

```
C:\W7>set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_25
```

```
C:\W7>set PATH=C:\FOP1.1;%JAVA_HOME%\bin;%PATH%
```

```
C:\W7>fop -fo hello.fo -pdf hello.pdf
```

# XSL-FO 문서

- XSL-FO 마크업과 콘텐츠로 구성된 XML 문서
- 확장자 .fo를 가짐
- fo:root 루트 엘리먼트로 문서를 구성
- fo:layout-master-set 컨테이너 엘리먼트를 이용해 문서 전체 구조를 기술
- fo:layout-master-set의 자식 엘리먼트 fo:simple-page-master에 대해 fo:page-sequence 엘리먼트를 1대 1로 구성
- fo:simple-page-master의 자식 엘리먼트 fo:region-body에 대해 fo:flow 엘리먼트를 1대 1로 구성해 페이지 콘텐츠를 기술



# XSL-FO 문서 구조

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

```
<fo:root xmlns:fo=http://www.w3.org/1999/XSL/Format>
```

루트 엘리먼트 노드

```
<fo:layout-master-set>
```

```
<fo:simple-page-master master-name="A4-portrait" page-width="29.7cm" page-height="21.0cm" margin="2cm">
```

```
<fo:region-body/>
```

```
</fo:simple-page-master>
```

```
</fo:layout-master-set>
```

```
<fo:page-sequence master-reference="A4-portrait">
```

```
<fo:flow flow-name="xsl-region-body">
```

```
<fo:block font-size="14pt" text-align="center">
```

페이지 콘텐츠

```
</fo:block>
```

```
</fo:flow>
```

```
</fo:page-sequence>
```

```
</fo:root>
```

최종 결과물의 전체 구조에 대한 템플릿을  
하나 이상의 simple-page-master를 기술  
(페이지 너비/높이, 여백, 표지, 목차 등)

최종 결과물의 페이지 콘텐츠를 기술  
하나 이상의 fo:block을 이용해 페이지  
컨텐츠를 기술

# fo:block 요소와 스타일 적용

```
<fo:page-sequence master-reference="A4-portrait">
  <fo:flow flow-name="xsl-region-body">
    <fo:block font-size="24pt" space-after="1cm">
      제목
    </fo:block>
    <fo:block>
      <fo:block font-size="14pt" space-after="0.5cm">
        내용 1
      </fo:block>
      <fo:block font-size="14pt" space-after="0.5cm">
        내용 2
      </fo:block>
    </fo:block>
  </fo:flow>
</fo:page-sequence>
```

# fo:block 요소와 이미지 적용

```
<fo:page-sequence master-reference="A4-portrait">
  <fo:flow flow-name="xsl-region-body">
    <fo:block font-size="24pt" space-after="1cm">
      제목
    </fo:block>
    <fo:block text-align="center" space-after="1cm">
      <fo:external-graphic src="url('fop.jpg')" content-height="6cm"/>
    </fo:block>
    <fo:block font-size="14pt" space-after="0.5cm">
      내용
    </fo:block>
    ...
  </fo:flow>
</fo:page-sequence>
```

# 페이지 템플릿 머릿글 생성

...

```
<fo:layout-master-set>
```

```
  <fo:simple-page-master master-name="A4-portrait" page-width="29.7cm" page-height="21.0cm" margin="2cm">
```

```
    <fo:region-body margin="2cm"/>
```

```
    <fo:region-before extent="2cm"/>
```

```
  </fo:simple-page-master>
```

```
</fo:layout-master-set>
```

```
<fo:page-sequence master-reference="A4-portrait">
```

```
  <fo:static-content flow-name="xsl-region-before">
```

```
    <fo:block font-size="18pt" text-align="center" border-bottom-width="medium"
```

```
      border-bottom-style="solid" margin="1cm">
```

```
      헤더 내용
```

```
    </fo:block>
```

```
  </fo:static-content>
```

...

# 페이지 템플릿 바닥글 생성

...

<fo:layout-master-set>

<fo:simple-page-master master-name="A4-portrait" page-width="29.7cm" page-height="21.0cm" margin="2cm">

<fo:region-body margin="2cm"/>

...

<fo:region-after extent="2cm"/>

</fo:simple-page-master>

</fo:layout-master-set>

<fo:page-sequence master-reference="A4-portrait">

...

<fo:static-content flow-name="xsl-region-after">

<fo:block font-size="9pt" text-align="right" margin="1cm">

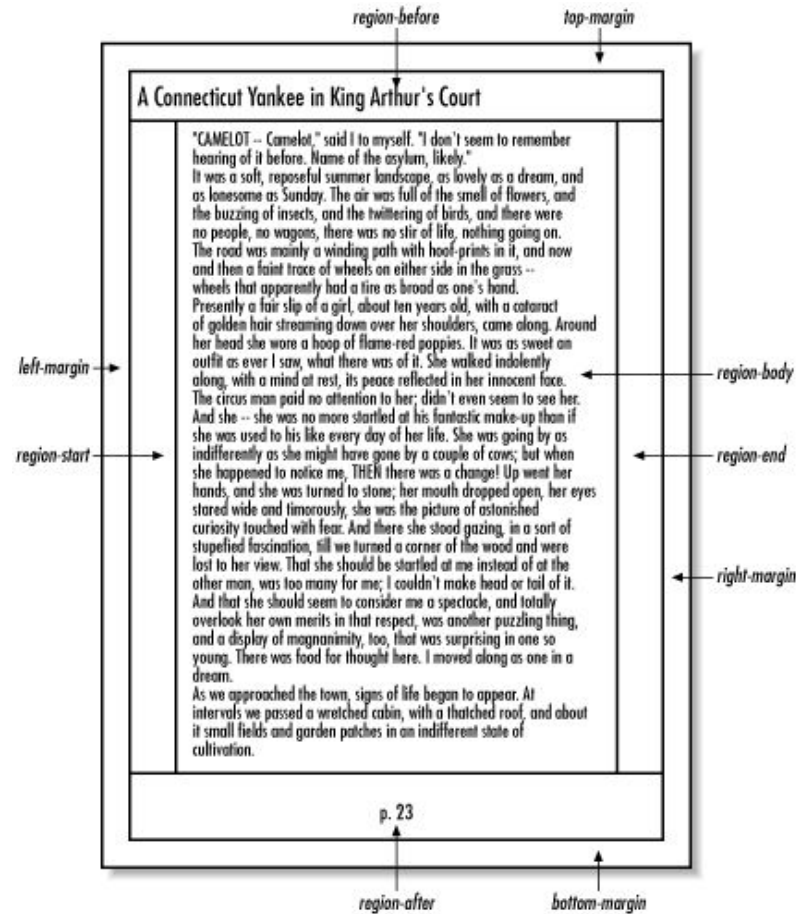
페이지 <fo:page-number/>

</fo:block>

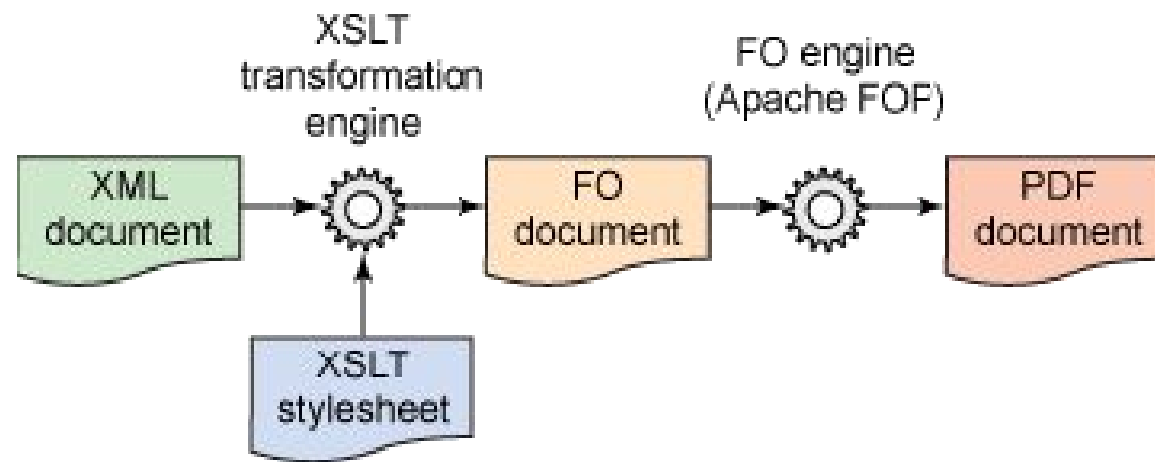
</fo:static-content>

...

# Region



# XSLT를 이용한 XSL-FO 변환



# XML 문서 – name.xml

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



# XSLT 문서 – name2fo.xsl

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="/">
    <fo:root>
      <fo:layout-master-set>
        <fo:simple-page-master
          master-name="A4-portrait"
          page-height="29.7cm" page-width="21.0cm"
          margin="2cm">
          <fo:region-body/>
        </fo:simple-page-master>
      </fo:layout-master-set>
```

```
    <fo:page-sequence master-reference="A4-portrait">
      <fo:flow flow-name="xsl-region-body">
        <fo:block>
          Hello, <xsl:value-of select="//name"/>!
        </fo:block>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>
</xsl:stylesheet>
```

# Apache™ FOP 환경 설정 – 한글 폰트 매트릭스 생성

```
C:\W7>set JAVA_HOME=C:\Program Files\Java\jdk1.8.0_25
```

```
C:\W7>set PATH=C:\FOP1.1;%JAVA_HOME%\bin;%PATH%
```

```
C:\W7>copy C:\Windows\Fonts\gulim.ttc .
```

```
C:\FOP1.1>java -cp build\fop.jar;lib\avalon-framework-4.2.0.jar;lib\commons-logging-1.0.4.jar;lib\commons-io-1.3.1.jar;lib\xmlgraphics-commons-1.5.jar  
org.apache.fop.fonts.apps.TTFReader -ttcname "Gulim" C:\W7\gulim.ttc  
C:\W7\gulim.xml
```

# Apache™ FOP 환경 설정 – mycfg.xml

```
...
<renderers>
  <renderer mime="application/pdf">
    ...
    <font metrics-url="gulim.xml" kerning="yes"
      embed-url="gulim.ttc">
      <font-triplet name="serif" style="normal" weight="normal"/>
      <font-triplet name="serif" style="normal" weight="bold"/>
      <font-triplet name="serif" style="italic" weight="normal"/>
      <font-triplet name="serif" style="italic" weight="bold"/>
      <font-triplet name="sans-serif" style="normal"
        weight="normal"/>
      <font-triplet name="sans-serif" style="normal"
        weight="bold"/>
      <font-triplet name="sans-serif" style="italic"
        weight="normal"/>
```

```
    <font-triplet name="sans-serif" style="italic"
      weight="bold"/>
    <font-triplet name="monospace" style="normal"
      weight="normal"/>
    <font-triplet name="monospace" style="normal"
      weight="bold"/>
    <font-triplet name="monospace" style="italic"
      weight="normal"/>
    <font-triplet name="monospace" style="italic"
      weight="bold"/>
  </font>
  ...
</renderer>
...
</renderers>
```

# Apache™ FOP 실행

```
C:\W7>fop -c mycfg.xml -xml name.xml -xsl name2fo.xsl -pdf name.pdf
```

# 학습정리

- XSL-FO 문서는 fo 확장자를 가지는 XML 문서로 인쇄용 문서를 생성할 목적으로 만들어졌다.
- XSL-FO 문서에서 fo:layout-master-set 엘리먼트 안에 문서 전체 구조를 기술한다.
- XSL-FO 문서에서 페이지 콘텐츠 추가 시 fo:region-body 엘리먼트를 사용한다.
- XSL-FO 문서에서 페이지 머리글 추가 시 fo:region-before 엘리먼트를 사용한다.
- XSL-FO 문서에서 페이지 바닥글 추가 시 fo:region-after 엘리먼트를 사용한다.
- XSLT 스타일시트를 이용해 XSL-FO 문서를 생성한 후 XSL-FO 프로세서로 인쇄용문서(PDF 등)을 만들 수 있다.