

# XSLT와 XML 문서 변환

(2주차)

# 학습개요

- 학습 목표

- XSLT가 무엇인지 설명할 수 있다.
- XSLT를 이용한 XML 문서 변환 프로세스를 설명할 수 있다.
- XSLT 스타일시트를 작성할 수 있다.

- 학습 내용

- XSLT와 변환 프로세스
- XSLT 스타일시트
- XSLT 스타일시트 실습

# XSLT

- eXtensible Style Language Transformations
- XSLT를 이용해 XML 문서에 대한 변환 결과 문서를 생성
  - 엘리먼트, 애트리뷰트의 추가/삭제
  - 특정 정보 추출
  - 특정 기준으로 정렬 등
- XSLT는 특정 정보를 찾기 위해 XPath를 사용함
- XML 문서 변환을 위해 원본 XML 문서와 XSLT 스타일시트, XSLT 프로세서가 필요함
- XSLT 프로세서는 원본 XML 문서를 분석하여 노드 트리를 구성한 후, XSLT 스타일시트를 적용해 문서 변환 결과를 생성함

# XSLT 프로세서와 노드 트리 1

파일명: contacts.xml

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

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

```
<contacts>
```

```
<contact>
```

```
<name language="Korean">이순신</name>
```

```
<name language="English">Soonshin Lee</name>
```

```
<phone>010-1234-1111</phone>
```

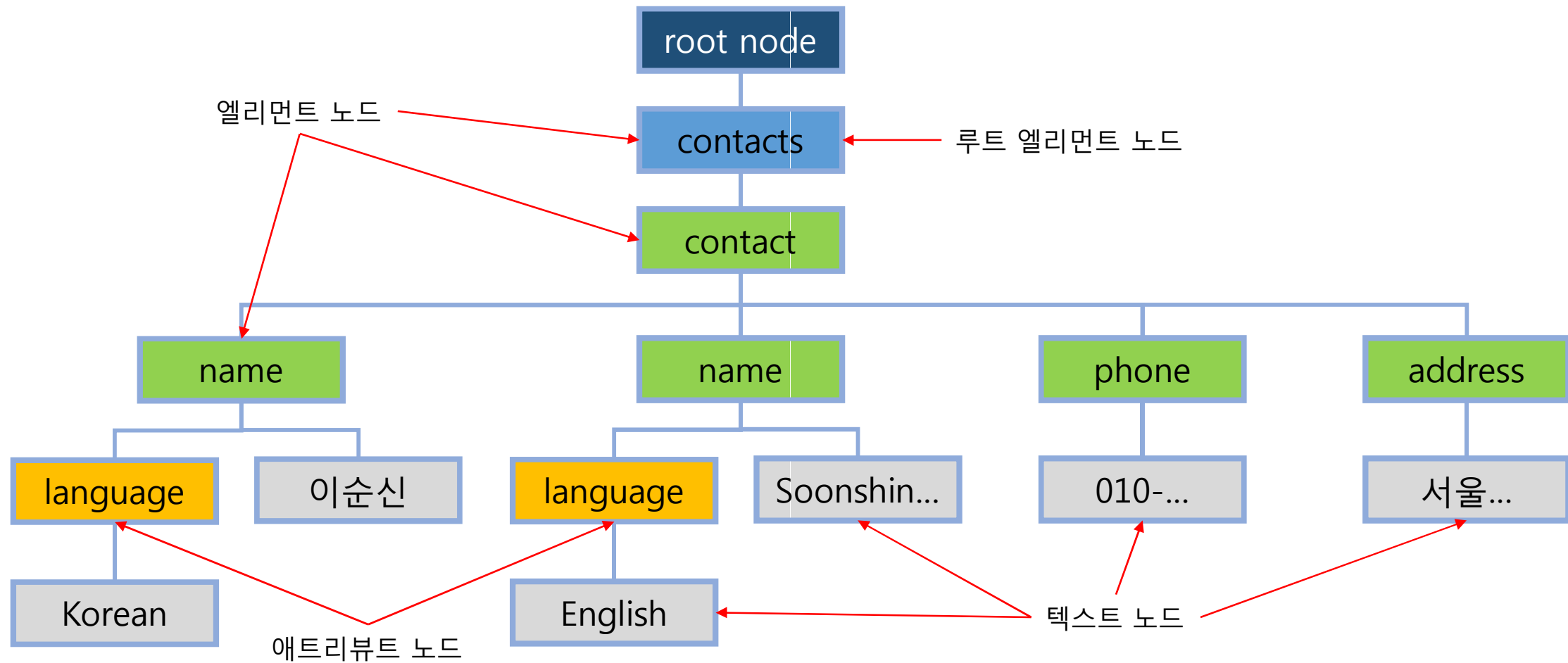
```
<address>서울특별시 송파구 신천동</address>
```

```
</contact>
```

```
</contacts>
```

XML 스타일시트 프로세싱 명령(PI)

# XSLT 프로세서와 노드 트리 2



# XSLT 스타일시트 적용과 결과 생성

파일명: contacts.xml

스타일시트에 대한 XML 네임스페이스를 기술하고  
접두어 xsi를 선언

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

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

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

루트 템플릿 정의

```
<html>
<head> <title>학생 정보</title> </head>
<body>
<h1>학생 정보</h1>
<strong> <xsl:value-of select="contacts/contact/name[@language='Korean']"/> </strong>
(<xsl:value-of select="contacts/contact/name[@language='English']"/>) <br/>
<xsl:value-of select="contacts/contact/phone"/> <br/>
<xsl:value-of select="contacts/contact/address"/>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

## 학생 정보

이순신 (Soonshin Lee)  
010-1234-1111  
서울특별시 송파구 신천



# XSLT 스타일시트

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

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

```
</xsl:stylesheet>
```

- XSLT 스타일시트 역시 XML 문서이며 잘 구성된(well-formed) 문서여야 함
- xsl:stylesheet 태그에서 스타일시트에 대한 XML 네임스페이스를 기술하고 접두어로 xsl을 선언함

# 루트 템플릿 생성

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

- XSLT 프로세서는 XSLT 스타일시트에서 루트 템플릿 (<xsl:template match="/">...</xsl:template>)을 가장 먼저 찾아 변환을 시작함
- 루트 템플릿은 XML 문서의 루트 노드에 적용될 규칙을 정의함



# HTML 출력

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

<xsl:template match="/">

  <html>
  <head> <title>학생 정보</title> </head>
  ...
  </html>

</xsl:template>
```

- XSLT 프로세서의 출력 방식을 xml, html, text로 지정할 수 있음 (생략될 경우 xml이 적용됨)
- 루트 템플릿의 출력으로 HTML을 추가

# 특정 XML 노드의 내용 출력하기

```
<xsl:template match="/">  
  
<html>...<body> <strong>  
<xsl:value-of select="contacts/contact/name[@language='Korean']"/>  
</strong> </body>...</html>  
  
</xsl:template>
```

- `<xsl:value-of select=" expression" />`
- *expression*은 XPath를 사용하여 XML 원본 문서에서 출력되어 질 내용을 갖는 노드들을 식별함
- 애트리뷰트를 이용한 식별에 `[@attribute='value']`를 사용

# 여러 XML 노드들을 반복 접근하기 1

- `<xsl:for-each select="expression">...</xsl:for-each>`
- *expression*은 XPath를 사용하여 XML 원본 문서에서 반복해서 접근할 노드들을 식별함
- `<xsl:value-of>` 사용 시 경로에서 `<xsl:for-each>`에 기술된 경로는 생략함

# 여러 XML 노드들을 반복 접근하기 2

contacts2.xml

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<?xml-stylesheet type="text/xsl" href="contacts2.xsl"?>
```

**<contacts>**

**<contact>**

```
<name language="Korean">이순신</name>  
<name language="English">Soonshin Lee</name>  
<phone>010-1234-1111</phone>  
<address>서울특별시 송파구 신천동</address>
```

**</contact>**

**<contact>**

```
<name language="Korean">홍길동</name>  
<name language="English">Gildong Hong</name>  
<phone>010-2345-2345</phone>  
<address>서울특별시 강남구 역삼동</address>
```

**</contact>**

**<contact>**

```
<name language="Korean">김유신</name>  
<phone>010-3456-4545</phone>  
<address>서울특별시 종로구 삼청동</address>
```

**</contact>**

**</contacts>**

# 여러 XML 노드들을 반복 접근하기 3

contacts2.xsl

```
...  
<xsl:template match="/">  
  <html>  
    <head>  
      <title>학생 정보2</title>  
    </head>  
    <body>  
      <h1>학생 정보2</h1>  
      <xsl:for-each select="contacts/contact">  
        <xsl:value-of select="name[@language='Korean']"/>  
        (<xsl:value-of select="phone"/>)<br/>  
      </xsl:for-each>  
    </body>  
  </html>  
</xsl:template>  
...
```

## 학생 정보2

이순신 (010-1234-1111)  
홍길동 (010-2345-2345)  
김유신 (010-3456-4545)



# 조건을 만족하는 XML 노드 처리하기 1

- `<xsl:if test="expression">...</xsl:if>`
- *expression*은 XPath를 사용하여 XML 원본 문서에서 조건을 만족하는 노드들을 식별함

# 조건을 만족하는 XML 노드 처리하기 2

contacts3.xsl

```
...  
<xsl:template match="/">  
<html>  
...  
<xsl:for-each select="contacts/contact">  
<strong>  
<xsl:value-of select="name[@language='Korean']"/>  
</strong>  
<xsl:if test="name[@language!='Korean']">  
( <xsl:value-of select="name[@language!='Korean']"/> )|  
</xsl:if>  
<xsl:value-of select="phone"/> <br/>  
</xsl:for-each>  
...  
</html>  
</xsl:template>  
...
```

## 학생 정보3

이순신 (Soonshin Lee) | 010-1234-1111  
홍길동 (Gildong Hong) | 010-2345-2345  
김유신 010-3456-4545



# 선택 조건으로 XML 노드 처리하기 1

- `<xsl:choose>`
  - `<xsl:when test=" expression`
  - `<xsl:when test=" expression`
  - `<xsl:otherwise>...</xsl:otherwise>``</xsl:choose>`
- *expression*은 XPath를 사용하여 XML 원본 문서에서 노드가 조건을 만족하는지 판단함
- 각 조건은 상호배타적임



# 선택 조건으로 XML 노드 처리하기 2

contacts4.xsl

```
...  
<xsl:template match="/">  
<html>  
...  
<xsl:for-each select="contacts/contact">  
<strong>  
<xsl:value-of select="name[@language='Korean']"/>  
</strong>  
<xsl:choose>  
<xsl:when test="age >= 18">Y</xsl:when>  
<xsl:otherwise>N</xsl:otherwise>  
</xsl:choose>  
<br/>  
</xsl:for-each>  
...  
</html>  
</xsl:template>  
...
```

학생 정보4

이순신 Y  
홍길동 Y  
김유신 N



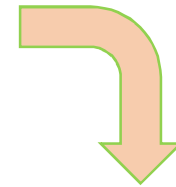
# XML 노드 정렬하기 1

- `<xsl:sort select="criteria" order="ascending|descending" data-type="text|number"/>`
- *criteria*는 노드들을 정렬하기 위한 기준 노드를 기술
- `<xsl:sort>`는 내포 구조로 여러 노드를 이용한 정렬이 가능

# XML 노드 정렬하기 2

contacts5.xsl

```
...  
<xsl:template match="/">  
<html>  
...  
<xsl:for-each select="contacts/contact">  
  <xsl:sort select="age" order="ascending" data-type="number"/>  
  <strong>  
    <xsl:value-of select="name[@language='Korean']"/>  
  </strong>  
  (<xsl:value-of select="age"/>)  
  <br/>  
</xsl:for-each>  
...  
</html>  
</xsl:template>  
...
```



학생 정보5

김유신 (16)  
홍길동 (20)  
이순신 (22)

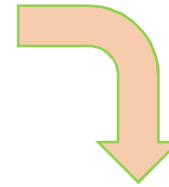
# 애트리뷰트 생성하기 1

- `<xsl:attribute name="att_name">...</xsl:attribute>`
- *att\_name*은 엘리먼트가 가져야하는 애트리뷰트 이름을 기술
- `<xsl:attribute>`의 내용으로 애트리뷰트 값을 기술

# 애트리뷰트 생성하기 2

contacts5.xsl

```
...
<xsl:template match="/">
<html>
...
<xsl:for-each select="contacts/contact">
  <a>
    <xsl:attribute name="href">
      #<xsl:value-of select="name[@language='Korean']"/>
    </xsl:attribute>
    <strong>
      <xsl:value-of select="name[@language='Korean']"/>
    </strong>
    </a>
    (<xsl:value-of select="age"/>)<br/>
  </xsl:for-each>
...
</html>
</xsl:template>
...
```



학생 정보6

이순신 (22)  
홍길동 (20)  
김유신 (16)

# 학습정리

- XSLT를 이용해 XML 문서에 대한 변환 결과 문서를 생성
- XSLT는 특정 정보를 찾기 위해 XPath를 사용함
- XSLT 프로세서는 원본 XML 문서를 분석하여 노드 트리를 구성한 후, XSLT 스타일시트를 적용해 문서 변환 결과를 생성함
- XSLT 스타일시트 역시 XML 문서이며 잘 구성된(well-formed) 문서여야 함
- XSLT 스타일시트의 루트 템플릿은 XML 문서의 루트 노드에 적용될 규칙을 정의함