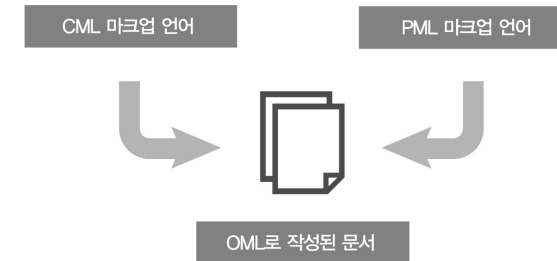


## Markup 언어의 혼용

- ◆ XML 문서는 하나의 markup 언어로 작성하는 것이 일반적이나, 여러 개의 markup 언어들을 이용해서 작성하는 것도 가능함



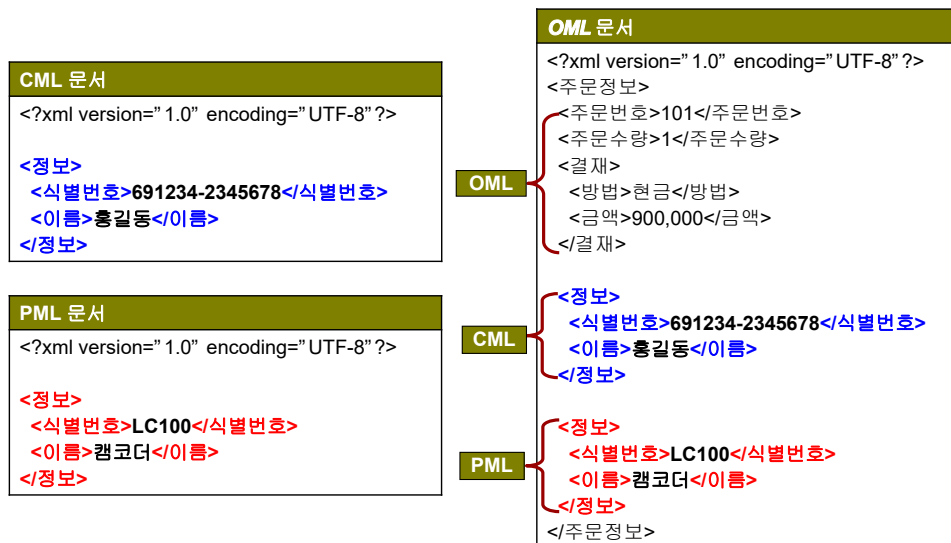
- CML: Customer Markup Language
- PML: Product Markup Language
- OML: Order Markup Language

1

## 5. XML Namespaces

### 엘리먼트 이름 충돌(collision)

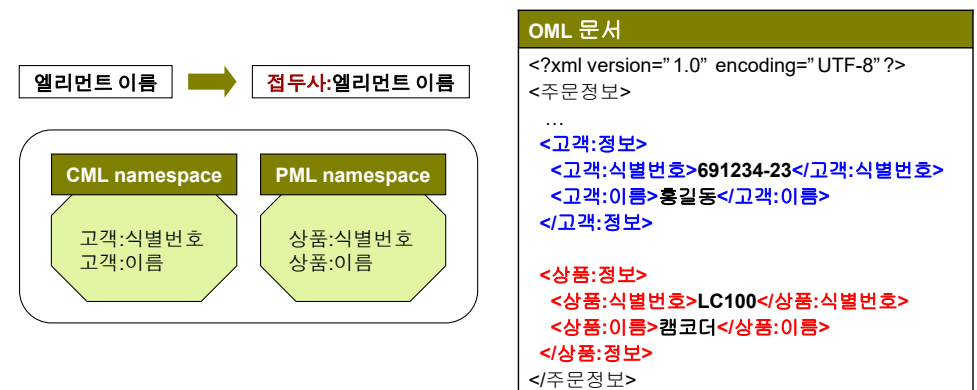
- ◆ 여러 언어에서 정의된 같은 이름의 엘리먼트들을 사용할 때 이름 충돌 발생 가능



### Namespace 개념

- ◆ 해결방안: namespace를 이용해서 각 언어에서 정의되는 이름들의 집합을 구분
  - 이름 앞에 namespace를 나타내는 접두사(prefix)를 붙여 사용

*Java 클래스, package*



3

## Namespace 이름

- ◆ Markup 언어의 namespace 이름은 그 언어를 개발할 때 정의
  - XML Schema 문서에서 "targetNamespace" 속성으로 정의
- ◆ Namespace 이름은 URI(Uniform Resource Identifier) 형태로 정의
  - URL(Uniform Resource Locator)
    - 예: "http://www.w3.org/1999/xhtml" (XHTML)
    - "http://xmlns.com/foaf/0.1/" (FOAF)
    - "http://purl.org/rss/1.0/" (RSS)
  - URN(Uniform Resource Name)
    - 예: "urn:isbn:0-486-27557-4" (ISBN)
    - "urn:issn:0167-6423" (ISSN)
- ◆ XML 문서에서 namespace 선언
  - Markup 언어가 namespace 이름을 갖고 있을 경우, XML 문서에서 namespace에 관한 속성 선언을 해야 함
    - Schema Location 선언: namespace 이름과 XML Schema 문서의 URL을 선언
    - Namespace 이름에 대한 접두사(prefix) 선언

4

## Namespace 이름

- ◆ Markup 언어의 스키마 문서에서 target namespace 정의 예

- bml.xsd (XML schema) *DTD 역할*

```
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.bookforum.or.kr/bml/1.0" ... >

  <xs:element name="booklist" type="booklistType"/>
  <xs:element name="book" type="bookType"/> ....
</xs:schema>
```

namespace 정의

- ◆ XML 문서에서 namespace 사용 예

```
<bml:booklist
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bmlforum.or.kr/bml/1.0
    http://www.bmlforum.or.kr/bml/1.0/bml.xsd"
  xmlns:bml="http://www.bmlforum.or.kr/bml/1.0">

  <bml:book> ... </bml:book> ...
</bml:booklist>
```

schema location 선언

namespace prefix 선언

5

## Namespace 사용 방법

- ◆ XML 문서에서 namespace 이름을 대신할 접두사를 선언
  - 일반적으로 루트 엘리먼트의 시작 태그 안에서 선언함
- ◆ 접두사는 XML 이름 작성 규칙을 따름
  - 이름은 문자나 "\_"로 시작할 수 있음
  - 두 번째 문자부터는 숫자, "\_", "-", "." 도 가능 ("." 문자는 사용할 수 없음)
  - 공백을 포함할 수 없음
- ◆ 각 엘리먼트(local name) 이름 앞에 접두사(prefix)를 붙여서 사용
  - 해당 엘리먼트 이름이 그 namespace에 속한다는 것을 의미함
  - XML parser는 prefix를 포함한 "Qualified Name"으로 엘리먼트를 인식(식별)

```
<!-- namespace prefix 선언 -->
<root-element xmlns:prefix="namespace 이름" ... >

  <!-- namespace 사용 -->
  <prefix:local-name> ... </prefix:local-name>

</root-element>
```

Qualified Name (QName)

6

## Namespace 사용 방법

- ◆ 예 1: OML 문서 (OML에는 namespace가 정의되지 않았다고 가정)
  - CML과 PML도 함께 사용

```
<주문정보 xmlns:고객="http://www.example.com/2003/CustomerML"
  xmlns:상품="http://www.example.com/2003/ProductML" ... >
  <주문번호>101</주문번호>
  <주문수량>1</주문수량>
  <결제>
    <방법>현금</방법>
    <금액>900,000</금액>
  </결제>
  <고객:정보>
    <고객:식별번호>691234-2345678</고객:식별번호>
    <고객:이름>홍길동</고객:이름>
  </고객:정보>
  <상품:정보>
    <상품:식별번호>LC100</상품:식별번호>
    <상품:이름>캠코더</상품:이름>
  </상품:정보>
</주문정보>
```

7

## Namespace 사용 방법

- ◆ 예 2: OML 문서 (OML에도 namespace가 정의된 경우)
  - 주의: namespace 선언이 포함된 루트 엘리먼트의 이름도 OML의 namespace에 속하므로 namespace prefix를 사용해야 함

```
<주문:주문정보 xmlns:주문="http://www.example.com/2003/OrderML"
  xmlns:고객="http://www.example.com/2003/CustomerML"
  xmlns:상품="http://www.example.com/2003/ProductML" ...>
  <주문:주문번호>101</주문:주문번호>
  <주문:주문수량>1</주문:주문수량>
  <주문:결제>
    <주문:방법>현금</주문:방법>
    <주문:금액>900,000</주문:금액>
  </주문:결제>
  <고객:정보>
    <고객:식별번호>691234-2345678</고객:식별번호>
    <고객:이름>홍길동</고객:이름>
  </고객:정보>
  <상품:정보> ...
</주문:주문정보>
```

8

## Namespace 사용 방법

- ◆ 예 3: 두 개의 언어(namespace)를 사용한 문서

```
<st:students xmlns:st="http://www.dongduk.ac.kr/studentML"
  xmlns:pf="http://www.w3.org/1999/xhtml" ... >
  <st:student>
    <st:sid>100</st:sid>
    <st:name>홍길동</st:name>
    <st:age>30</st:age>
    <st:profile>
      <pf:html>
        <pf:head>
          <pf:title>나의 소개</pf:title>
        </pf:head>
        <pf:body>학력: ... 경력: ... 특기: ... </pf:body>
      </pf:html>
    </st:profile>
  </st:student>
</st:students>
```

pf  
언어  
오류!

XHTML 이용

9

## Namespace 사용 방법

- ◆ 예 4: Spring Framework 설정 파일 (4개의 namespace 사용)

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:context="http://www.springframework.org/schema/context"
  xmlns:tx="http://www.springframework.org/schema/tx"
  xmlns:mybatis="http://mybatis.org/schema/mybatis-spring"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd
    http://www.springframework.org/schema/tx
    http://www.springframework.org/schema/tx/spring-tx.xsd
    http://mybatis.org/schema/mybatis-spring
    http://mybatis.org/schema/mybatis-spring-1.2.xsd">

  <!-- data source 설정 -->
  <bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource">
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" />
    <property name="url" value="jdbc:oracle:thin:@202.20.119.117:1521:orcl" />
    <property name="username" value="spring" />
    <property name="password" value="spring" />
  </bean>
```

상세 작성X  
→ 에디터 사용

## Namespace 사용 방법

```
<!-- MyBatis SqlSessionFactoryBean 생성 -->
<bean id="sqlSessionFactoryBean" class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="typeAliasesPackage" value="com.example.model" />
  <property name="dataSource" ref="dataSource" />
  <property name="configLocation" value="classpath:/mybatis-config.xml" />
  <property name="mapperLocations">
    <array>
      <value>classpath*/com/example/repository/mapper/**/*.xml</value>
    </array>
  </property>
</bean>

<!-- MyBatis MapperScannerConfigurer 설정:
  Mapper interface scan 및 mapper bean 자동 생성 -->
<mybatis:scan base-package="com.example.repository.mapper"/>

<!-- 트랜잭션 관리자 설정 -->
<bean id="transactionManager"
  class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource" />
</bean>

<!-- Annotation을 이용한 트랜잭션 설정 -->
<tx:annotation-driven transaction-manager="transactionManager" />

<!-- @Repository 애노테이션을 이용한 자동 빈 등록 -->
<context:component-scan base-package="com.example.repository" />
</beans>
```

11

## Namespace 사용 방법

### ◆ Namespace 적용 범위

- Namespace prefix가 선언된 엘리먼트를 포함해서, 그 엘리먼트의 자손 엘리먼트들에 대해서만 적용 가능
- 예

XML 문서 : c5\_0801.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bml:booklist>
  <bml:book
    xmlns:bml="http://www.example.com/2003/BookML">
    <bml:title>시인과 도둑</bml:title>
    <bml:author>이문열</bml:author>
  </bml:book>

  <bml:book>
    <bml:title>봉순이 언니</bml:title>
    <bml:author>공지영</bml:author>
  </bml:book>
</bml:booklist>
```

오류: bml namespace 적용 불가능

bml namespace 적용 가능

오류: bml namespace 적용 불가능  
bml은 연속 문항성에 붙여서

12

## Default Namespaces

### ◆ Default namespace: 접두사를 생략하고 선언 및 사용하는 namespace

- 접두사를 생략한 모든 이름들은 default namespace에 포함된 것으로 간주
- 예

```
<?xml version="1.0" encoding="UTF-8"?>
<상품정보 xmlns="http://www.example.com/2003/ProductML">
  <상품번호>LC100</상품번호>
  <상품명>캠코더</상품명>
  <제조사>LG전자</제조사>
</상품정보>
```

접두사 생략

### ◆ Default namespace 선언은 여러 개의 markup 언어를 혼용해서 작성하는 XML 문서에서도 사용 가능

- 문서에서 가장 많이 사용되는 markup 언어에 대해 default namespace를 선언하고, 다른 markup 언어들에 대해서는 각각 namespace prefix를 선언함

13

## Default Namespaces

### ◆ 예: OML 문서 (OML의 namespace를 default namespace로 선언)

```
<주문정보 xmlns="http://www.example.com/2003/OrderML"
  xmlns:고객="http://www.example.com/2003/CustomerML"
  xmlns:상품="http://www.example.com/2003/ProductML">
  <주문번호>101</주문번호>
  <주문수량>1</주문수량>
  <결제>
    <방법>현금</방법>
    <금액>900000</금액>
  </결제>
  <고객:정보>
    <고객:식별번호>691234-2345678</고객:식별번호>
    <고객:이름>홍길동</고객:이름>
  </고객:정보>
  <상품:정보>
    <상품:식별번호>LC100</상품:식별번호>
    <상품:이름>캠코더</상품:이름>
  </상품:정보>
</주문정보>
```

OML에 속한 엘리먼트들에 대해서는 namespace prefix를 생략 (p.8의 예와 비교)

14

## Attributes and Namespaces

### ◆ 속성(attribute)에 대한 namespace 적용

- 속성이 엘리먼트와 무관하게 (전역적으로) 정의되는 경우, namespace 접두사를 사용해서 특정 namespace에 포함됨을 명시함
- 속성이 특정 엘리먼트 내에 (지역적으로) 정의되는 경우에는 namespace 사용 불필요
- 속성에 대해서는 default namespace가 적용되지 않음

XML 문서 : c5\_1101.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bml:booklist xmlns:bml="http://www.example.com/2003/BookML">
  <bml:book bml:id="b2" bml:kind="소설">
    <bml:title>봉순이 언니</bml:title>
    <bml:author>공지영</bml:author>
  </bml:book>
</bml:booklist>
```

id, kind가 전역적으로 정의된 속성인 경우 namespace 접두사 사용

15