

XML Schema

(11주차)

학습개요

- 학습 목표

- XML Schema에 대해 설명할 수 있다.
- XML Schema와 XML 문서를 연결하는 방법을 설명할 수 있다.
- XML Schema의 심플 타입과 사용자 정의 심플 타입 활용 방법을 설명할 수 있다.
- XML Schema 문서를 작성하는 방법을 설명할 수 있다.

- 학습 내용

- XML Schema 소개
- 심플 타입
- 사용자 정의 심플 타입
- XML Schema 실습

XML Schema

- 2001년 W3C는 DTD의 단점을 보완하기 위해 XML Schema 언어를 개발
- <http://www.w3.org/XML/Schema>
- XML 구문을 사용
- XML 네임스페이스를 사용
- 엘리먼트와 엘리먼트의 관계, 애트리뷰트 등을 정의하여 유효한 XML 문서 구조를 정의
- 엘리먼트는 심플 타입이나 컴플렉스 타입으로 정의
- 심플 타입은 문자열, 숫자, 날짜 등의 값만 포함하며, 컴플렉스 타입은 자식 엘리먼트, 값 등을 가짐

XML Schema

- XML Schema는 XML 문서로 루트 엘리먼트는 schema이며, 네임스페이스는 W3C의 XML Schema 네임스페이스로 선언되며, ".xsd" 확장자를 가짐

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

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
...
```

```
</xs:schema>
```

-

XML Schema 위치 정의

- XML Schema와 XML 문서를 연결하려면 XML 문서의 루트 엘리먼트에 XML Schema 문서 위치를 선언

```
<root_element  
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="xsd.uri" >  
  ...  
</root_element>
```

-

XML Schema 예제

[note.xsd]

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="note">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="to" type="xs:string"/>
        <xs:element name="from" type="xs:string"/>
        <xs:element name="heading" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

[note.xml]

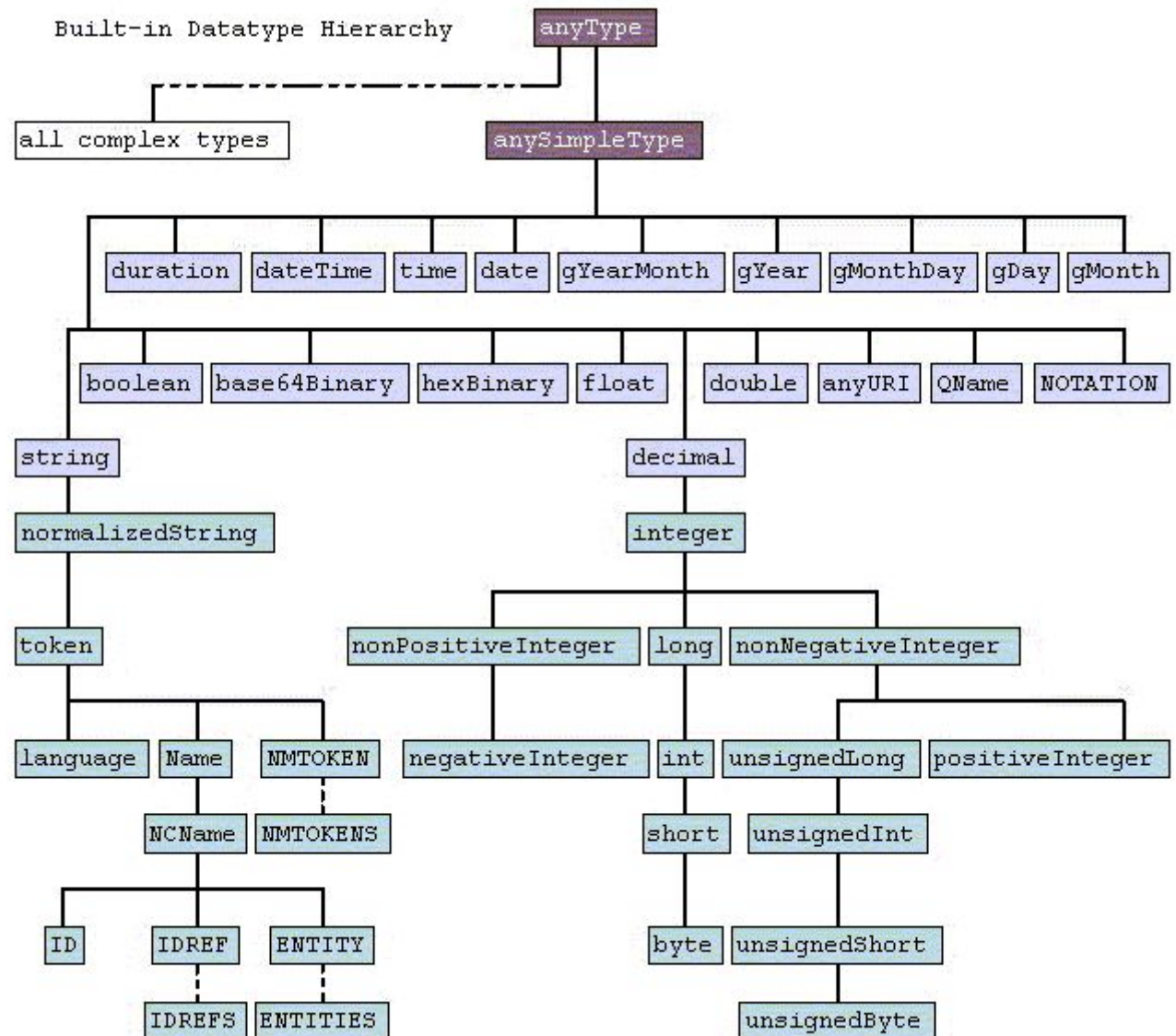
```
<?xml version="1.0" encoding="UTF-8"?>
<note xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="note.xsd" >
  <to>홍길동</to>
  <from>이순신</from>
  <heading>공지</heading>
  <body>퇴근 후 회식!!!</body>
</note>
```

심플 타입

- 심플 타입은 오직 값(문자열, 부울값, URL, 날짜, 시간, 숫자 등)만을 포함할 수 있음
- 빌트인 심플 타입은 xs: 로 정의되는 XML Schema 네임스페이스 접두사로 시작
- 자식 엘리먼트나 애트리뷰트를 포함할 수 없음

심플 타입

<http://www.w3.org/TR/xmlschema-2/built-in-datatypes>



xs:string

xsd:

```
<xs:element name="name" type="xs:string"/>
```

```
<xs:element name="age" type="xs:string"/>
```

xml:

```
<name>홍길동</name> <!-- 유효한 XML -->
```

```
<age>18세</age> <!-- 유효한 XML -->
```

```
<age>18</age> <!-- 유효한 XML -->
```

xs:date

xsd:

```
<xs:element name="birth" type="xs:date"/>
```

xml:

```
<birth>2001-05-24</birth> <!-- 유효한 XML -->
```

```
<age>May 24, 2014</age> <!-- 유효하지 않은 XML -->
```

format: YYYY-MM-DD

xs:time

xsd:

```
<xs:element name="arrival" type="xs:time"/>
```

xml:

```
<arrival>20:30:00</arrival> <!-- 유효한 XML -->
```

format: hh:mm:ss

xs:dateTime

xsd:

```
<xs:element name="event" type="xs:dateTime"/>
```

xml:

```
<event>2015-04-01T10:15:00</event> <!-- 유효한 XML -->
```

```
<event>2015-04-08T18:30:00+09:00</event> <!-- 유효한 XML -->
```

format: YYYY-MM-DDThh:mm:ss

xs:duration

xsd:

```
<xs:element name="period" type="xs:duration"/>
```

xml:

```
<period>P3D</period> <!-- 유효한 XML -->
```

```
<period>P2M5D</period> <!-- 유효한 XML -->
```

```
<period>P-3H</period> <!-- 유효한 XML -->
```

format: PnYnMnDTnHnMnS

xs:gYear

xsd:

```
<xs:element name="birth_year" type="xs:gYear"/>
```

xml:

```
<birth_year>2000</birth_year> <!-- 유효한 XML -->
```

format: YYYY

xs:gYearMonth

xsd:

```
<xs:element name="birth_year_month"  
type="xs:gYearMonth"/>
```

xml:

```
<birth_year_month>2000-12</birth_year_month> <!-- 유효한  
XML -->
```

format: YYYY-MM

xs:gMonth

xsd:

```
<xs:element name="birth_month" type="xs:gMonth"/>
```

xml:

```
<birth_month>--12</birth_month> <!-- 유효한 XML -->
```

format: --MM

xs:gMonthDay

xsd:

```
<xs:element name="birth_month_day" type="xs:gMonthDay"/>
```

xml:

```
<birth_month_day>--12-25</birth_month_day> <!-- 유효한  
XML -->
```

format: --MM-DD

xs:gDay

xsd:

```
<xs:element name="birth_day" type="xs:gDay"/>
```

xml:

```
<birth_day>---25</birth_month_day> <!-- 유효한 XML -->
```

format: ---DD

xs:decimal

xsd:

```
<xs:element name="random" type="xs:decimal"/>
```

xml:

```
<random>3.14</random> <!-- 유효한 XML -->
```

```
<random>0</random> <!-- 유효한 XML -->
```

```
<random>-10</random> <!-- 유효한 XML -->
```

xs:integer

xsd:

```
<xs:element name="age" type="xs:integer"/>
```

xml:

```
<age>18</age> <!-- 유효한 XML -->
```

```
<age>18세</age> <!-- 유효하지 않은 XML -->
```

```
<age>18.5</age> <!-- 유효하지 않은 XML -->
```

fixed 엘리먼트 값 설정하기

xsd:

```
<xs:element name="booktype" type="xs:string" fixed="epub"/>
```

xml:

```
<booktype>epub</booktype> <!-- 유효한 XML -->
```

```
<booktype></booktype> <!-- 유효한 XML -->
```

```
<booktype>mobi</booktype> <!-- 유효하지 않은 XML -->
```

default 엘리먼트 값 설정하기

xsd:

```
<xs:element name="booktype" type="xs:string"  
default="epub"/>
```

xml:

```
<booktype>epub</booktype> <!-- 유효한 XML -->
```

```
<booktype></booktype> <!-- 유효한 XML -->
```

```
<booktype>mobi</booktype> <!-- 유효한 XML -->
```

글로벌 심플 타입

xsd:

...

```
<xs:simpleType name="custom_type_name">  
  <xs:restriction base="foundation">  
    ... facet  
  </xs:restriction>  
</xs:simpleType>
```

...

```
<xs:element name="child_element" type="custom_type_name"/>
```

...

로컬 심플 타입

xsd:

...

```
<xs:element name="child_element">  
  <xs:simpleType>  
    <xs:restriction base="foundation">  
      ... facet  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>...
```

...

xs:restriction / xs:maxInclusive

xsd:

```
<xs:element name="score">  
  <xs:simpleType>  
    <xs:restriction base="xs:integer">  
      <xs:maxInclusive value="100">  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

xml:

```
<score>90</score> <!-- 유효한 XML -->  
<score>100</score> <!-- 유효한 XML -->  
<score>120</score> <!-- 유효하지 않은 XML -->
```

xs:restriction / xs:maxExclusive

xsd:

```
<xs:element name="score">  
  <xs:simpleType>  
    <xs:restriction base="xs:integer">  
      <xs:maxExclusive value="100">  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

xml:

```
<score>90</score> <!-- 유효한 XML -->  
<score>100</score> <!-- 유효하지 않은 XML -->
```

xs:restriction / xs:minInclusive

xsd:

```
<xs:element name="score">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="60">
        <xs:maxInclusive value="100">
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
```

xml:

```
<score>100</score> <!-- 유효한 XML -->
<score>60</score> <!-- 유효한 XML -->
<score>59</score> <!-- 유효하지 않은 XML -->
<score>101</score> <!-- 유효하지 않은 XML -->
```

xs:restriction / xs:enumeration

xsd:

```
<xs:element name="title">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="xml programming">
      <xs:enumeration value="javascript programming">
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

xml:

```
<title>xml programming</title> <!-- 유효한 XML -->
<title>javascript</title> <!-- 유효하지 않은 XML -->
<title>xml programming javascript programming</title> <!-- 유효하지 않은 XML -->
```

xs:restriction / xs:length

xsd:

```
<xs:element name="code">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="5">
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

xml:

```
<code>A0001</code> <!-- 유효한 XML -->
<code>B001</code> <!-- 유효하지 않은 XML -->
```

xs:restriction / xs:maxLength

xsd:

```
<xs:element name="sms">  
  <xs:simpleType>  
    <xs:restriction base="xs:string">  
      <xs:maxLength value="80">  
    </xs:restriction>  
  </xs:simpleType>  
</xs:element>
```

xml:

```
<sms> 최대 80문자까지 사용 가능합니다...</sms> <!-- 유효한 XML -->
```

xs:restriction / xs:pattern

xsd:

```
<xs:element name="isbn13">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="\d{3}-\d{10}">
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

xml:

```
<isbn13>123-0123456789</isbn13> <!-- 유효한 XML -->
<isbn13>123-4567890</isbn13> <!-- 유효하지 않은 XML -->
```

xs:restriction / xs:totalDigits, xs:fractionDigits

xsd:

```
<xs:element name="height">  
  <xs:simpleType>  
    <xs:restriction base="xs:decimal">  
      <xs:totalDigits value="5">  
        <xs:fractionDigits value="2">  
          </xs:restriction>  
        </xs:simpleType>  
      </xs:element>
```

xml:

```
<height>180.35</height> <!-- 유효한 XML -->  
<height>211.458</height> <!-- 유효하지 않은 XML -->
```


xs:list

xsd:

```
<xs:element name="departure">  
  <xs:simpleType>  
    <xs:list itemType="xs:dateTime"/>  
  </xs:simpleType>  
</xs:element>
```

xml:

```
<departure>  
  2015-04-01T09:30:00Z  
  2015-04-01T10:30:00Z  
  2015-04-01T11:30:00Z  
</departure> <!-- 유효한 XML -->
```

xs:union

xsd:

```
<xs:element name="isbn">  
  <xs:simpleType>  
    <xs:union memberTypes="isbn10 isbn13"/>  
  </xs:simpleType>  
</xs:element>
```

xml:

```
<isbn>123-0123456789</isbn> <!-- 유효한 XML -->  
<isbn>1234567890</isbn> <!-- 유효한 XML -->
```

학습정리

- XML Schema는 XML 구문을 사용해 XML 문서 구조를 정의합니다.
- XML Schema는 XML 네임스페이스를 사용합니다.
- 심플 타입은 문자열, 숫자, 날짜 등의 값만 포함하며, 자식 엘리먼트나 애트리뷰트를 포함할 수 없습니다.
- 사용자 정의 심플 타입은 `xs:restriction`, `xs:list`, `xs:union` 엘리먼트를 활용해 정의합니다.