XML Tutorial

출처 : w3schools.com/xml

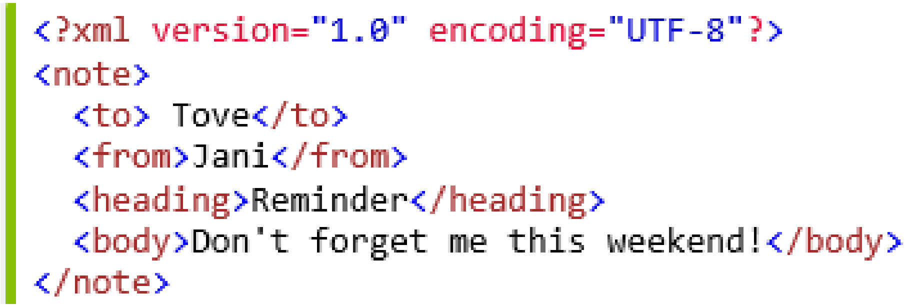
XML Basic

XML History

- 1970년대 중반, 찰스 F. 골드팔:
  - ✓ SGML (Standard Generalized Markup Language)
  - ✓ 데이터 표현의 국제적 기준 제시
- 1980년대 중반, 팀 버너스-리:
  - ✓ World Wide Web 고안, W3C 결성
- W3C 첫 번째 활동
  - ✓ HTML (Hypertext Markup Language) Spec.
  - ✓ 형식과 문서 텍스트의 레이아웃 설명 체계 제공
- 1990년대 후반, e-비즈니스 발전
  - ✓ 표준 비즈니스 중심의 데이터 표현 형식 요구.
  - ✓ For example, XML (Extensible Markup Language)
  - ✓ 메타 정보(자기 묘사가 가능한 태그) 제공

XML

- XML stands for eXtensible Markup Language
- XML is designed to describe data
- XML is a software- and hardware-independent tool for carrying information
- XML is easy to learn



```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to> Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Display this XML in Your Browser >>

Display Another >>   With Style >>

Introduction to XML

- XML was designed to describe data.
- HTML was designed to display data.

What You Should Already Know

Before you continue you should have a basic understanding of the following:

- HTML
- JavaScript

Introduction to XML

What is XML?

- XML stands for Extensible Markup Language
- XML is a markup language much like HTML
- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive
- XML is a W3C Recommendation

Introduction to XML

The Difference Between XML and HTML

- XML is not a replacement for HTML
- XML and HTML were designed with different goals:
    - ✓ XML was designed to describe data, with focus on what data is
    - ✓ HTML was designed to display data, with focus on how data looks
- HTML is about displaying information,

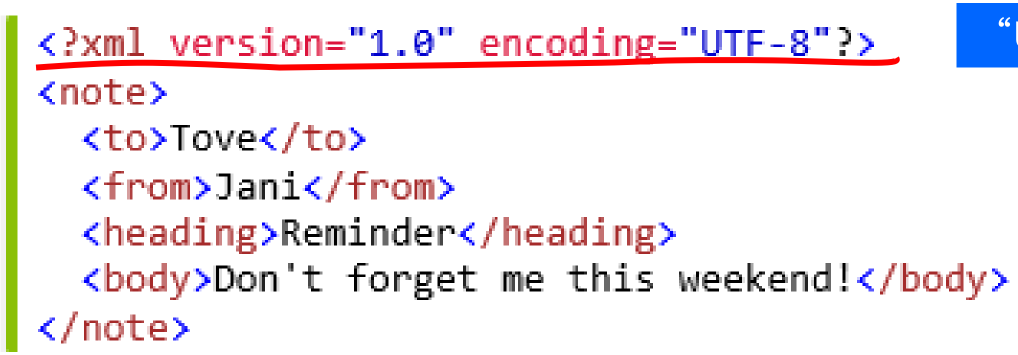
  while XML is about carrying information.

XML Tree

XML documents form a tree structure that starts at "root" and branches to "leaves".

An Example XML Document

- XML documents use a self-describing and simple syntax:



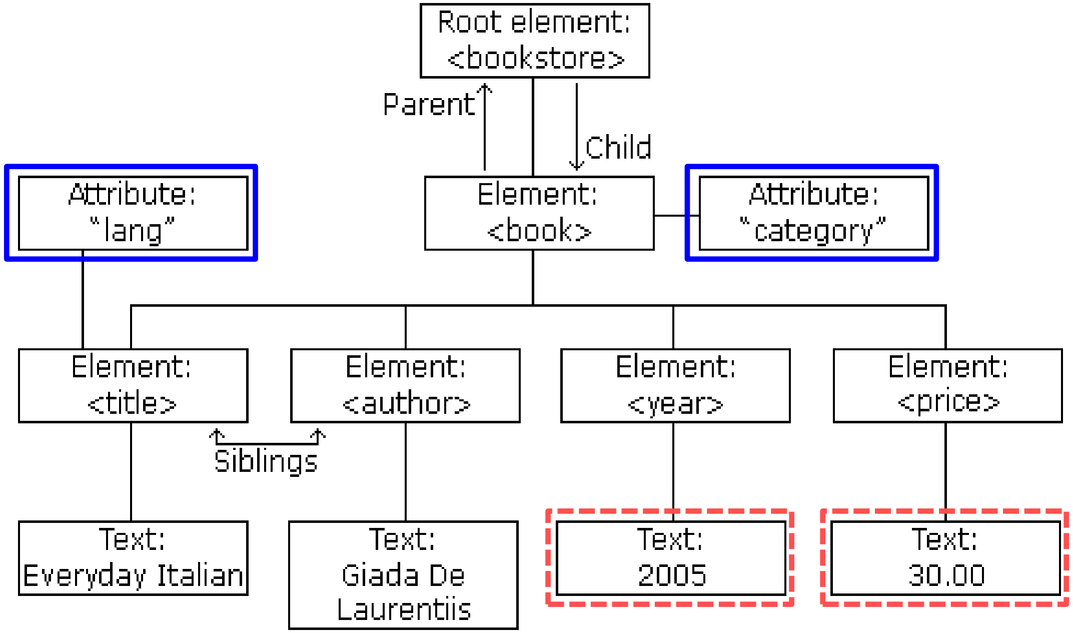
```xml
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

“UTF-8”

- The first line is the XML declaration

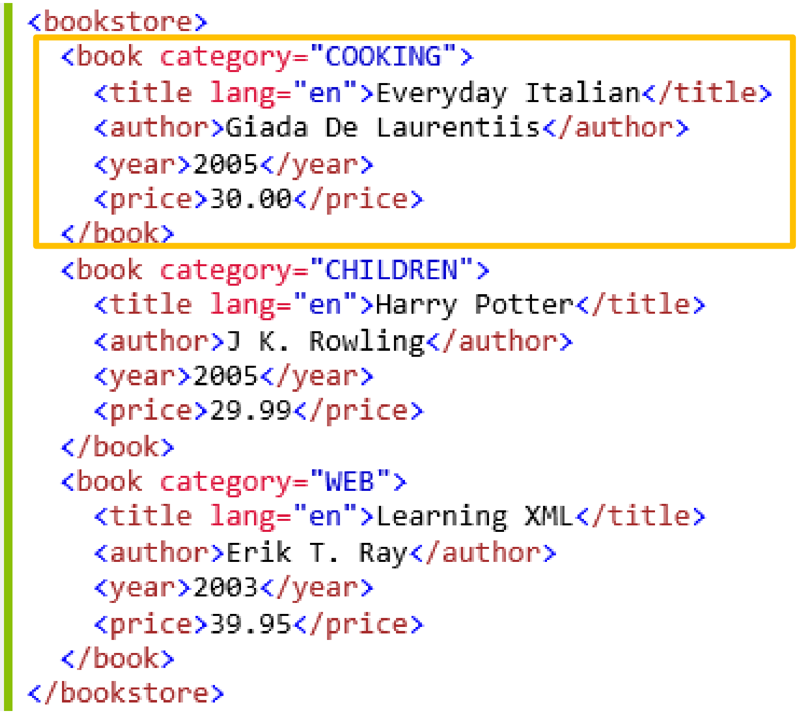
XML Tree

All elements can have text content and attributes



The image above represents one book in the XML below:

books_tree.xml

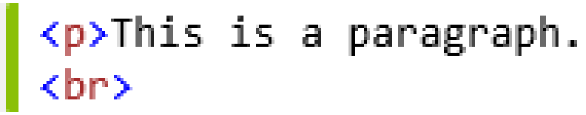


```xml
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="CHILDREN">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

XML Syntax (1/8)

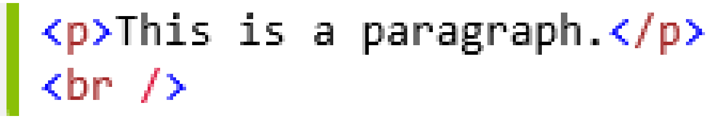
All XML Elements Must Have a Closing Tag

- In HTML, some elements do not have to have a closing tag:



```
<p>This is a paragraph.
<br>
```

- In XML, it is illegal to omit the closing tag
  - ✓ All elements must have a closing tag:



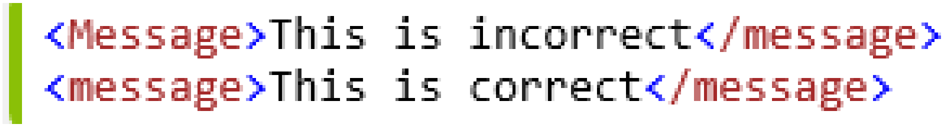
```
<p>This is a paragraph.</p>
<br />
```

XML Syntax (2/8)

XML Tags are Case Sensitive

- XML tags are case sensitive
  - ✓ The tag <Letter> is different from the tag <letter>

- Opening and closing tags must be written with the same case:

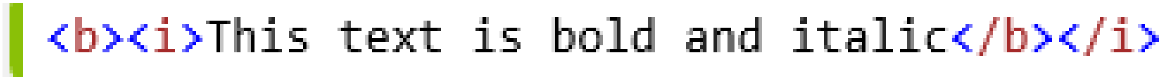


```
<Message>This is incorrect</message>
<message>This is correct</message>
```

XML Syntax (3/8)

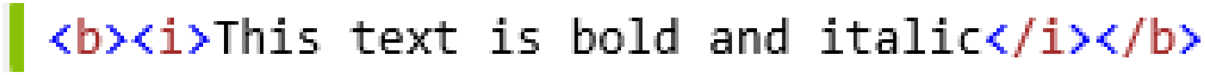
XML Elements Must be Properly Nested

- In HTML, you might see improperly nested elements:



```
<b><i>This text is bold and italic</b></i>
```

- In XML, all elements must be properly nested within each other:



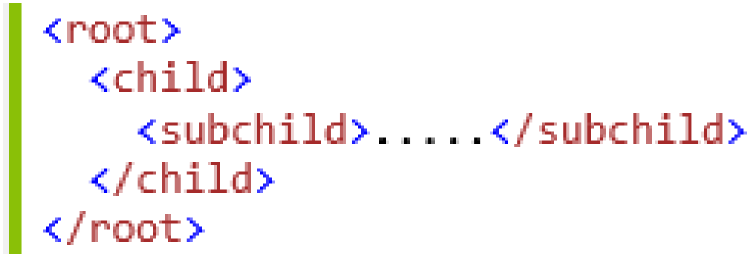
```
<b><i>This text is bold and italic</i></b>
```

- "Properly nested" simply means that since the <i> element is opened inside the <b> element, it must be closed inside the <b> element

XML Syntax (4/8)

XML Documents Must Have a Root Element

- XML documents must contain one element that is the parent of all other elements. This element is called the root element.

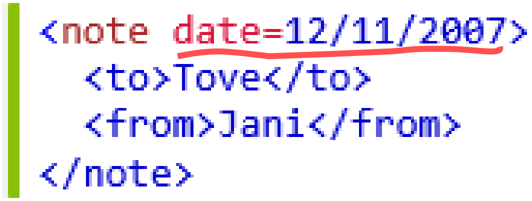


```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

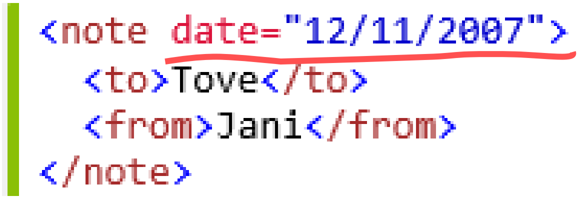
XML Syntax (5/8)

XML Attribute Values Must be Quoted

- XML elements can have attributes in name/value pairs just like in HTML
- In XML, attribute values must always be quoted
- Study the two XML documents below. The first one is incorrect, the second is correct:



```
<note date=12/11/2007>
  <to>Tove</to>
  <from>Jani</from>
</note>
```



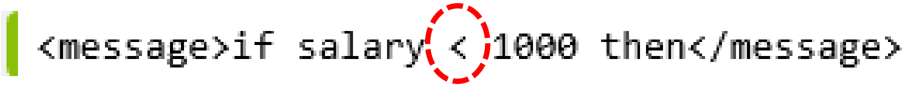
```
<note date="12/11/2007">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

XML Syntax (6/8)

Entity References (1/2)

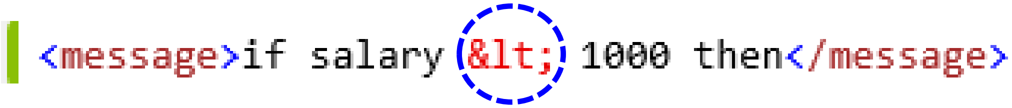
- Some characters have a special meaning in XML
- If you place a character like "<" inside an XML element, it will generate an error because the parser interprets it as the start of a new element

- This will generate an XML error:



```
<message>if salary < 1000 then</message>
```

- To avoid this error, replace the "<" character with an entity reference:

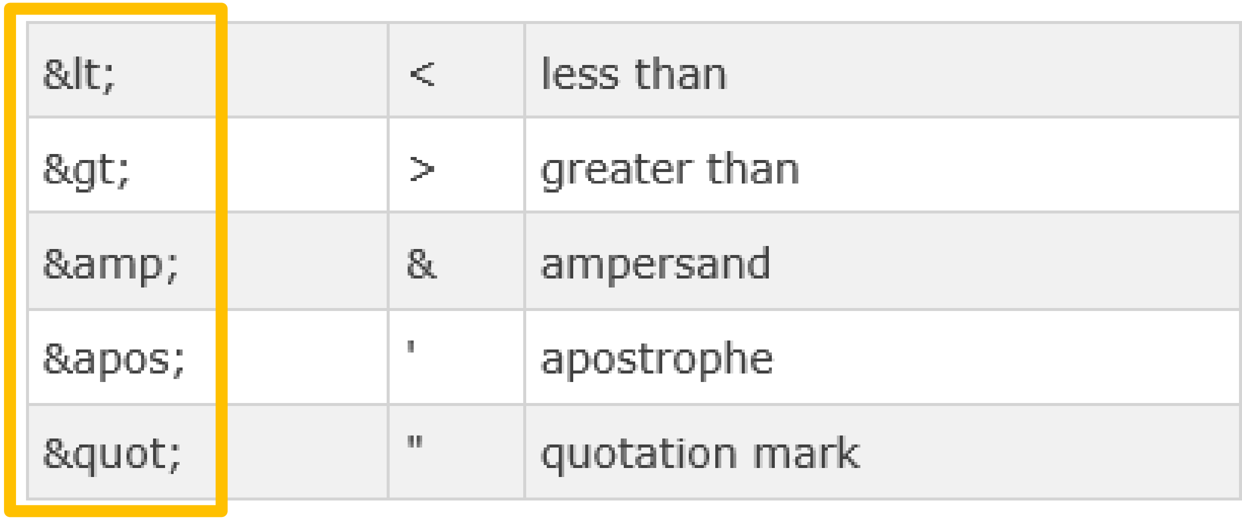


```
<message>if salary &lt; 1000 then</message>
```

XML Syntax (7/8)

Entity References (2/2)

- There are 5 predefined entity references in XML:



&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

XML Syntax (8/8)

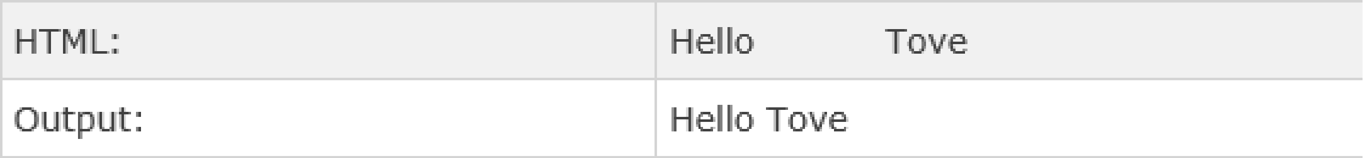
Comments in XML

- Syntax for writing comments in XML is similar to that of HTML

```
<!-- This is a comment -->
```

White-space is Preserved in XML

- HTML truncates multiple white-space characters to one single white-space:
  (HTML은 여러 개의 공백 문자를 하나의 공백 문자로 자릅니다.)



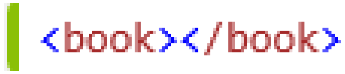
HTML:	Hello Tove
Output:	Hello Tove

- With XML, the white-space in a document is not truncated
  (XML에서도 문서의 공백이 잘리지 않습니다.)

XML Elements

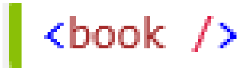
Empty XML Elements

- An alternative syntax can be used for XML elements with no content:
- Instead of writing a book element (with no content) like this:



```
<book></book>
```

- It can be written like this:

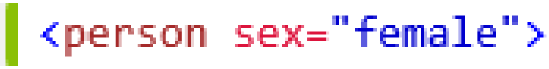


```
<book />
```

XML Attributes (1/2)

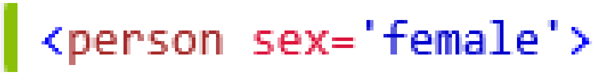
XML Attributes Must be Quoted

- Attribute values must always be quoted
  - ✓ Either single or double quotes can be used. For a person's sex, the person element can be written like this:



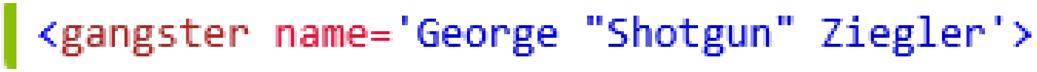
```
<person sex="female">
```

or like this:



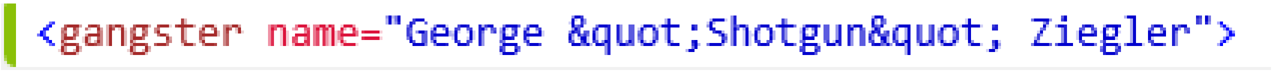
```
<person sex='female'>
```

- If the attribute value itself contains double quotes, you can use single quotes, like in this example:



```
<gangster name='George "Shotgun" Ziegler'>
```

- or you can use character entities:

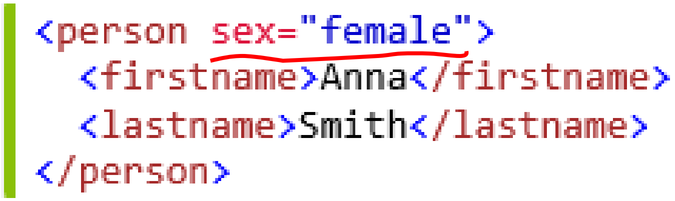


```
<gangster name="George &quot;Shotgun&quot; Ziegler">
```

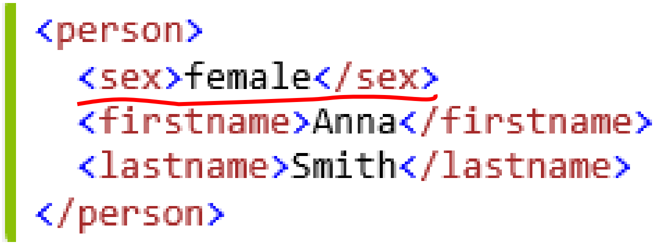
XML Attributes (2/2)

XML Elements vs. Attributes

- Take a look at these examples:



```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```



```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

Both examples provide the same information

- There are no rules about when to use attributes or when to use elements
- Attributes are handy in HTML
- In XML my advice is to avoid them. Use elements instead

XML Namespaces(1/2)

- **이름공간 또는 네임스페이스**(Namespace)

  개체를 구분할 수 있는 범위를 나타내는 말로 일반적으로 하나의 이름 공간에서는 하나의 이름이 단 하나의 개체만을 가리키게 된다.

- **Name Conflicts**

  In XML, element names are defined by the developer. This often results in a conflict when trying to mix XML documents from different XML applications.

  This XML carries HTML table information:

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

  This XML carries information about a table (a piece of furniture):

```
<table>
  <name>African Coffee Table</name>
  <width>80</width>
  <length>120</length>
</table>
```

  If these XML fragments were added together, there would be a name conflict. Both contain a <table> element, but the elements have different content and meaning.
  A user or an XML application will not know how to handle these differences.

참고 : https://www.w3schools.com/xml/xml_namespaces.asp

XML Namespaces(2/2)

- **Solving the Name Conflict Using a Prefix**

  Name conflicts in XML can easily be avoided using a name prefix.
  This XML carries information about an HTML table, and a piece of furniture:

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```

  In the example above, there will be no conflict because the two <table> elements have different names.

참고 : http://www.tcpschool.com/xml/xml_basic_namespace

<실습 순서>

참고 : www.w3schools.com/xml

<실습1> dom

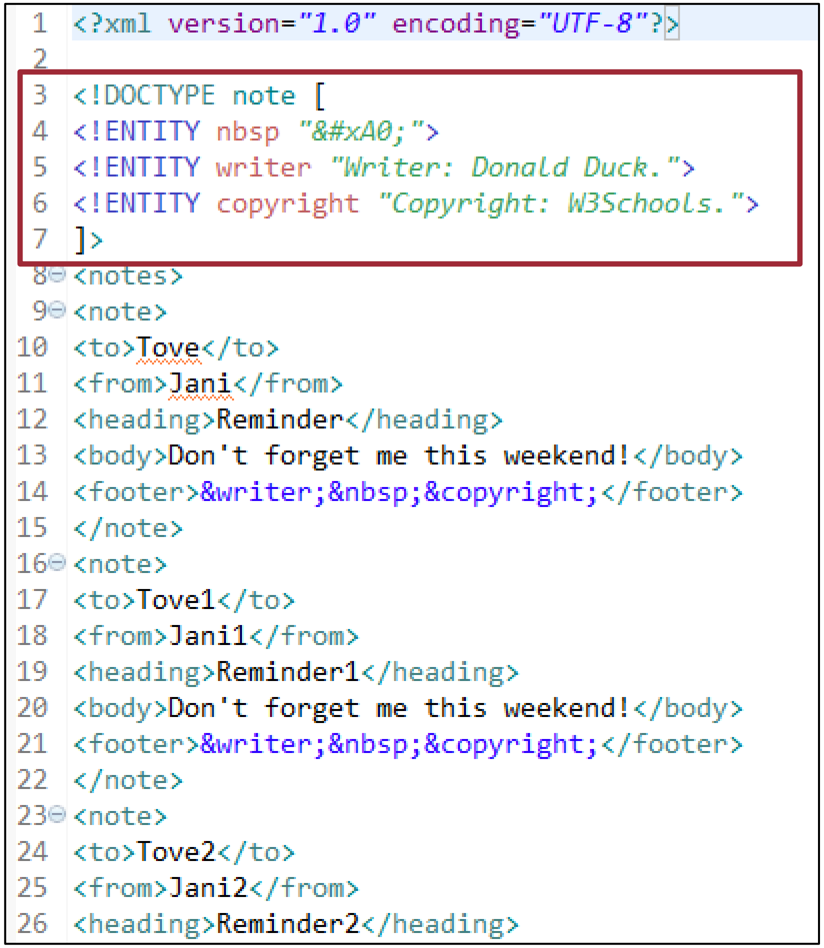
<실습2> xsl

<실습3> books(get, change, set(create), remove)

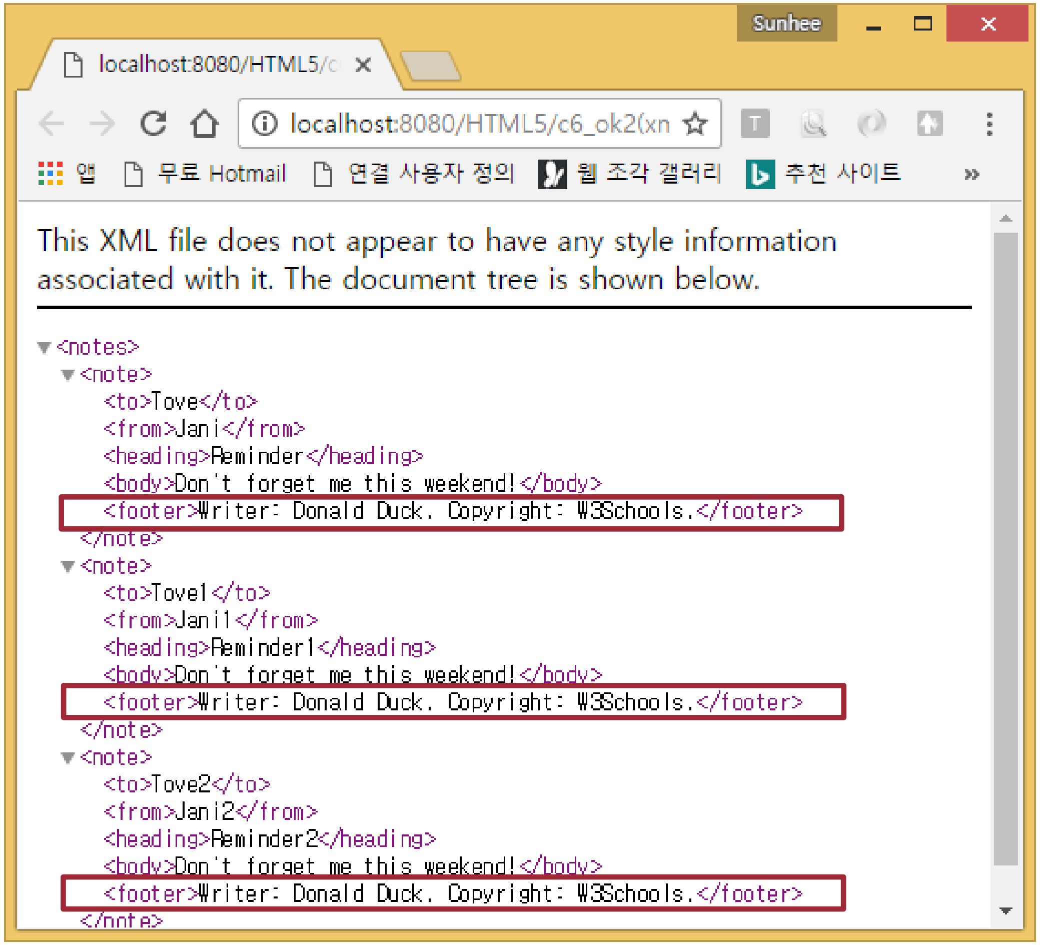
<실습4> cd_catalog

XML DTD – ENTITY의 활용 – note1.xml

- What is a DTD?
  1. DTD stands for Document Type Definition.
  2. A DTD defines the structure and the legal elements and attributes of an XML document.



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

<!DOCTYPE note [
<!ENTITY nbsp "&#xA0;">
<!ENTITY writer "Writer: Donald Duck.">
<!ENTITY copyright "Copyright: W3Schools.">
]>
<notes>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
<footer>&writer;&nbsp;&copyright;</footer>
</note>
<note>
<to>Tove1</to>
<from>Jani1</from>
<heading>Reminder1</heading>
<body>Don't forget me this weekend!</body>
<footer>&writer;&nbsp;&copyright;</footer>
</note>
<note>
<to>Tove2</to>
<from>Jani2</from>
<heading>Reminder2</heading>
```

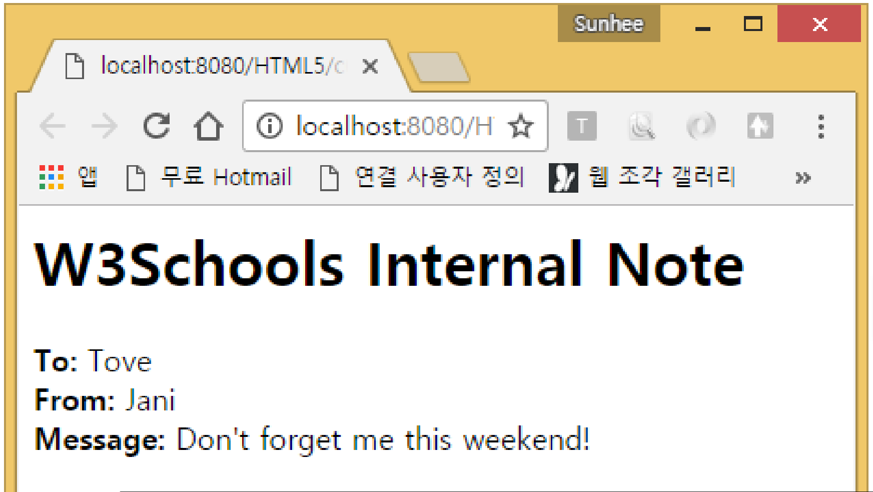


This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<notes>
  ▼<note>
     <to>Tove</to>
     <from>Jani</from>
     <heading>Reminder</heading>
     <body>Don't forget me this weekend!</body>
     <footer>Writer: Donald Duck. Copyright: W3Schools.</footer>
   </note>
  ▼<note>
     <to>Tove1</to>
     <from>Jani1</from>
     <heading>Reminder1</heading>
     <body>Don't forget me this weekend!</body>
     <footer>Writer: Donald Duck. Copyright: W3Schools.</footer>
   </note>
  ▼<note>
     <to>Tove2</to>
     <from>Jani2</from>
     <heading>Reminder2</heading>
     <body>Don't forget me this weekend!</body>
     <footer>Writer: Donald Duck. Copyright: W3Schools.</footer>
   </note>
```

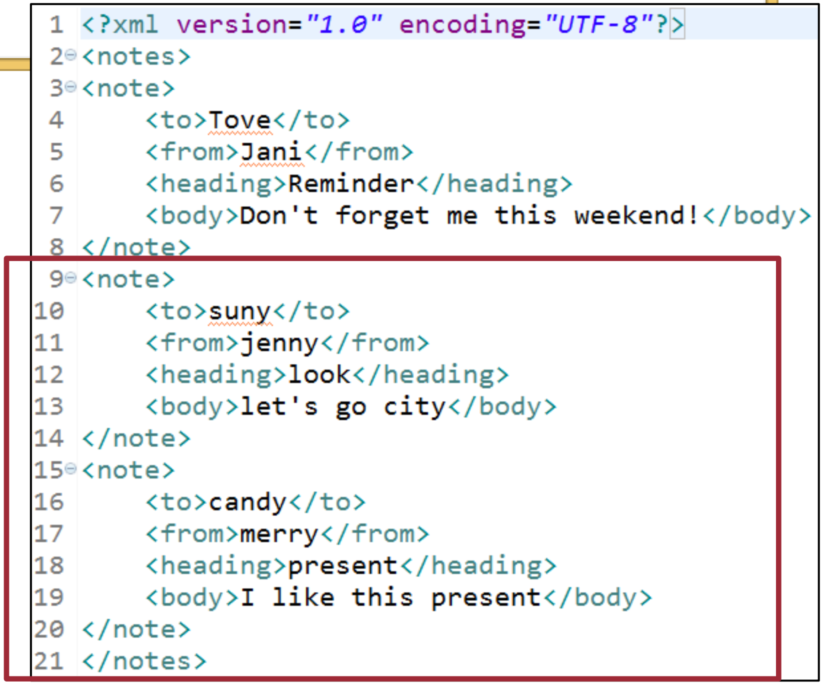
http://localhost:8000/c6_ex2(xml)/note1.xml

<실습1> Dom – xmldom.html



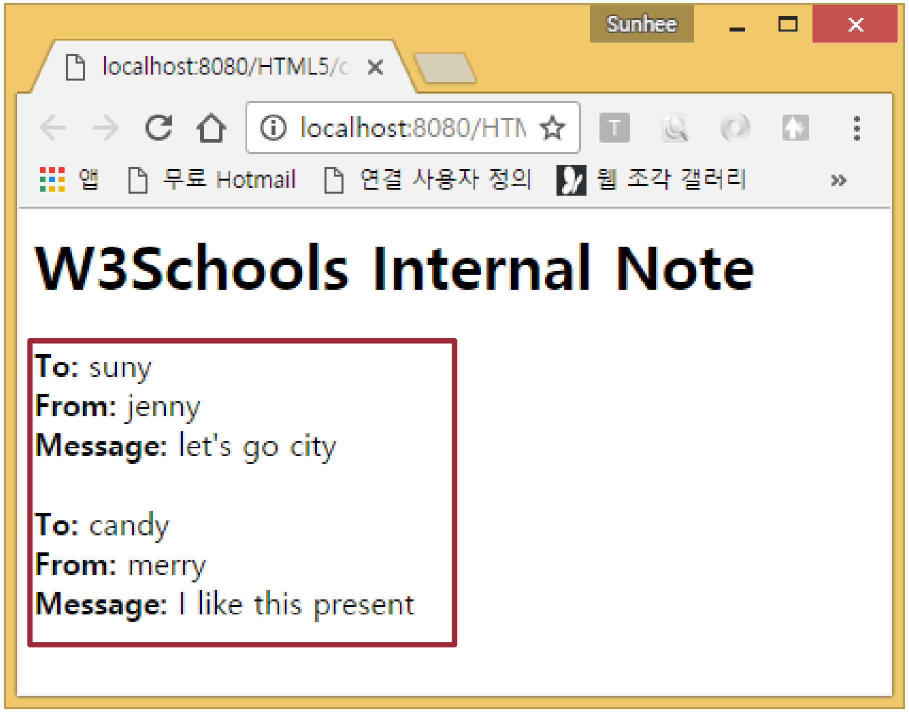
W3Schools Internal Note

To: Tove
From: Jani
Message: Don't forget me this weekend!



```
<?xml version="1.0" encoding="UTF-8"?>
<notes>
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
</note>
<note>
    <to>suny</to>
    <from>jenny</from>
    <heading>look</heading>
    <body>let's go city</body>
</note>
<note>
    <to>candy</to>
    <from>merry</from>
    <heading>present</heading>
    <body>I like this present</body>
</note>
</notes>
```

note.xml



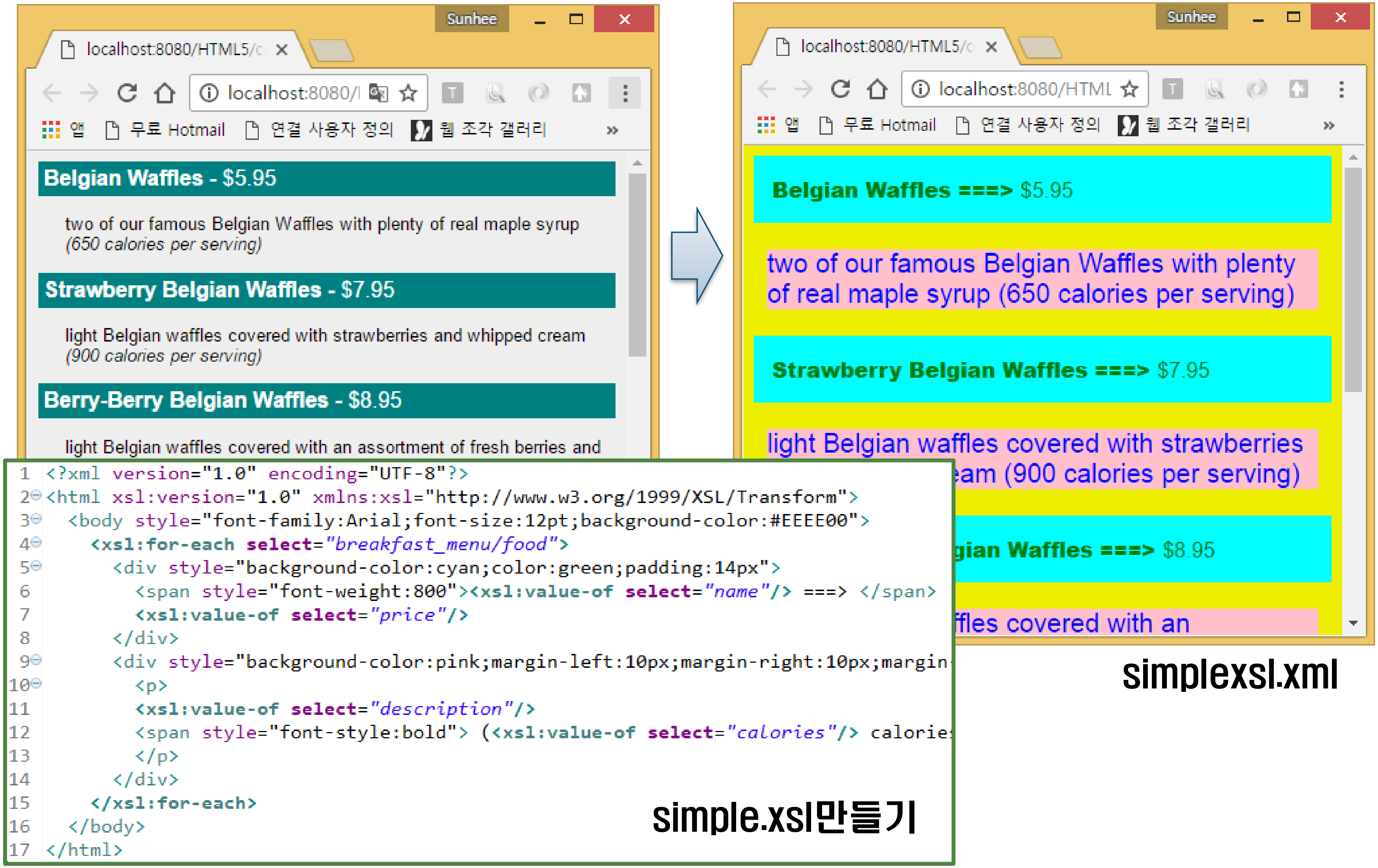
W3Schools Internal Note

To: suny
From: jenny
Message: let's go city

To: candy
From: merry
Message: I like this present

xmldom.html

<실습2> Xsl – simplexsl.xml



Belgian Waffles - $5.95

two of our famous Belgian Waffles with plenty of real maple syrup *(650 calories per serving)*

Strawberry Belgian Waffles - $7.95

light Belgian waffles covered with strawberries and whipped cream *(900 calories per serving)*

Berry-Berry Belgian Waffles - $8.95

light Belgian waffles covered with an assortment of fresh berries and

Belgian Waffles ===> $5.95

two of our famous Belgian Waffles with plenty of real maple syrup (650 calories per serving)

Strawberry Belgian Waffles ===> $7.95

light Belgian waffles covered with strawberries

eam (900 calories per serving)

gian Waffles ===> $8.95

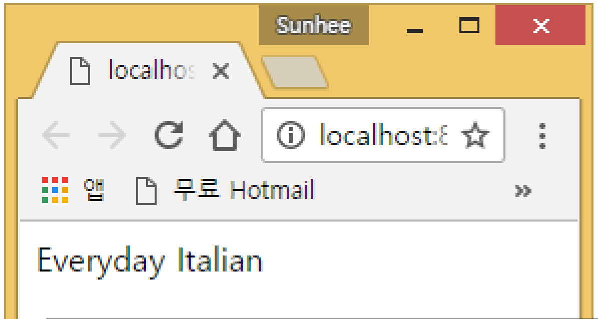
ffles covered with an

```
<?xml version="1.0" encoding="UTF-8"?>
<html xsl:version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <body style="font-family:Arial;font-size:12pt;background-color:#EEEE00">
    <xsl:for-each select="breakfast_menu/food">
      <div style="background-color:cyan;color:green;padding:14px">
        <span style="font-weight:800"><xsl:value-of select="name"/> ===> </span>
        <xsl:value-of select="price"/>
      </div>
      <div style="background-color:pink;margin-left:10px;margin-right:10px;margin
        <p>
        <xsl:value-of select="description"/>
        <span style="font-style:bold"> (<xsl:value-of select="calories"/> calorie
        </p>
      </div>
    </xsl:for-each>
  </body>
</html>
```

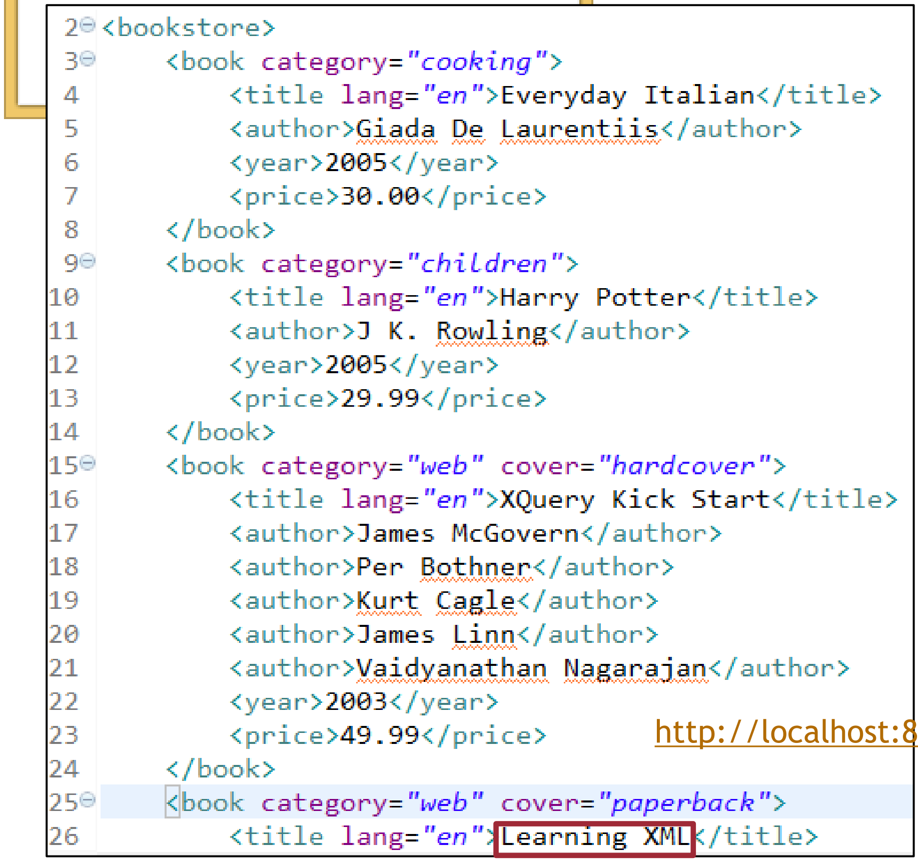
simple.xsl만들기

simplexsl.xml

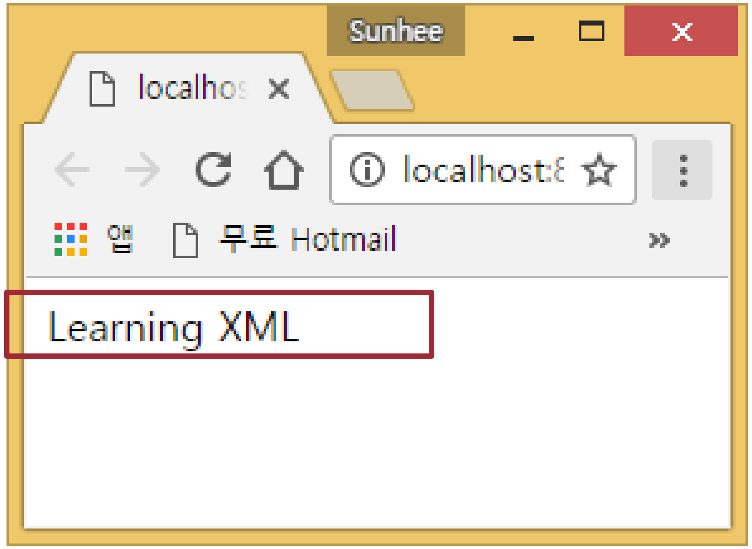
<실습3-1> books – advance_get_element1.html



Everyday Italian



```
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web" cover="hardcover">
        <title lang="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>2003</year>
        <price>49.99</price>
    </book>
    <book category="web" cover="paperback">
        <title lang="en">Learning XML</title>
```



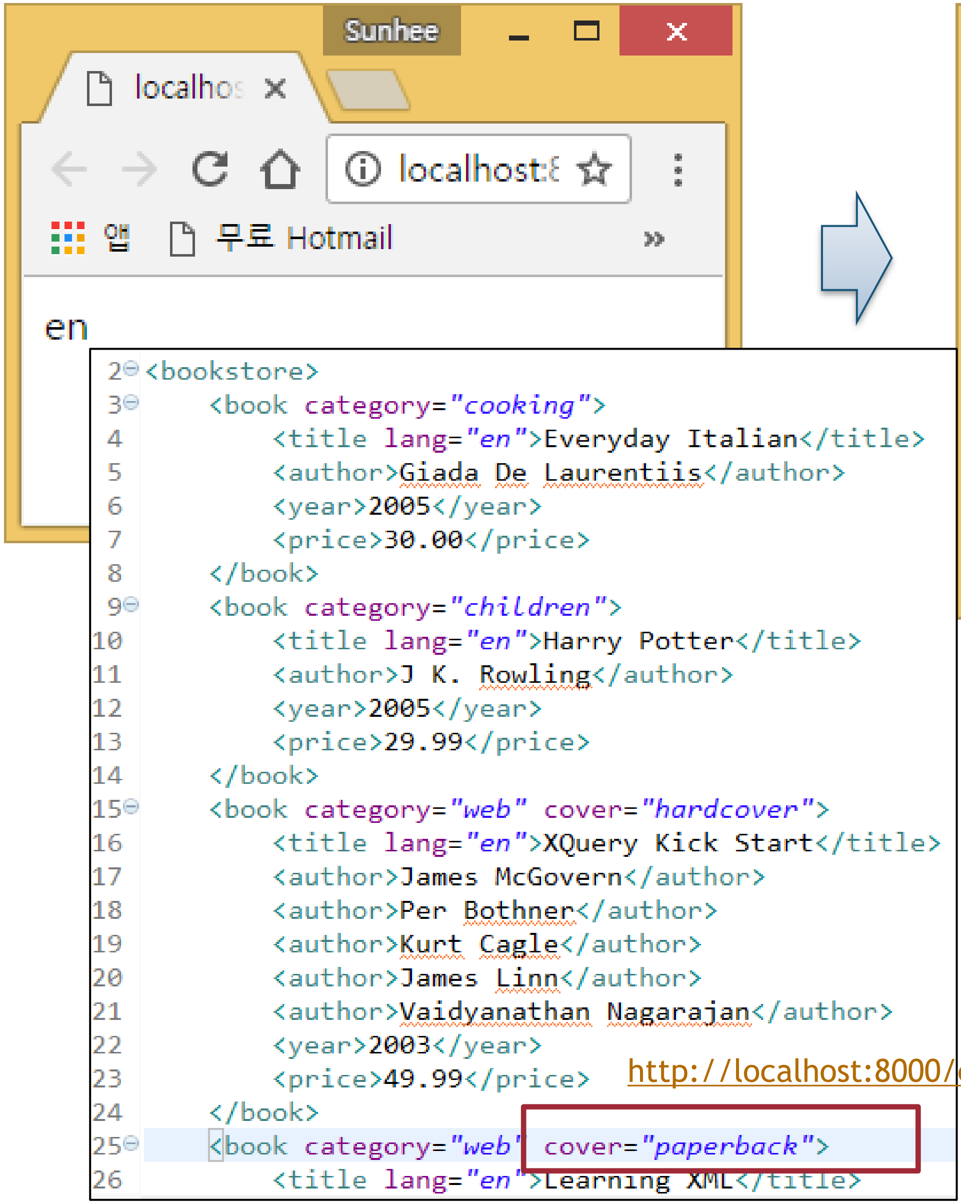
Learning XML

advance_get_element1.html

getElementsByTagName

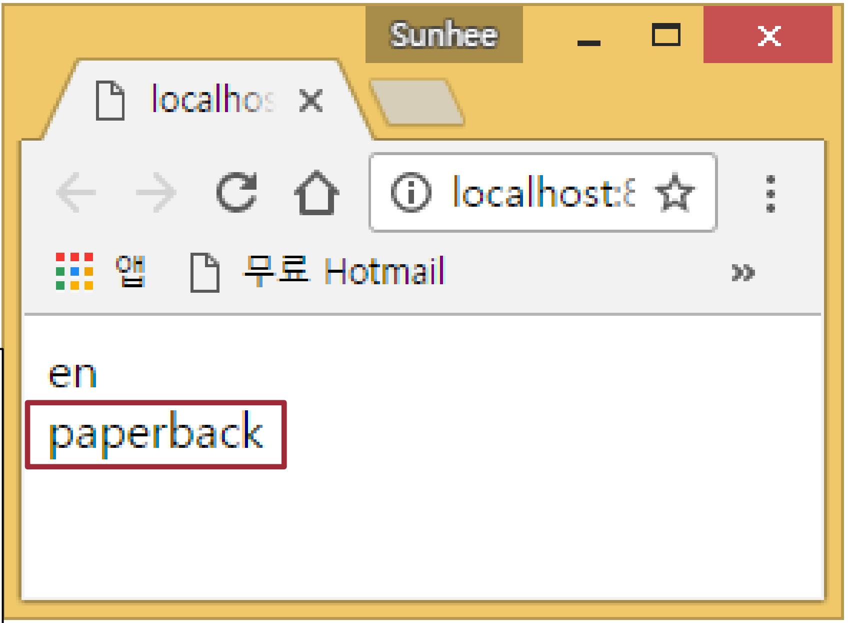
http://localhost:8000/c6_ex2(xml)/3.books/advance_get_element1.html

<실습3-2> books – advance_get_attribute2.html



en

```
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web" cover="hardcover">
        <title lang="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>2003</year>
        <price>49.99</price>
    </book>
    <book category="web" cover="paperback">
        <title lang="en">Learning XML</title>
```



en

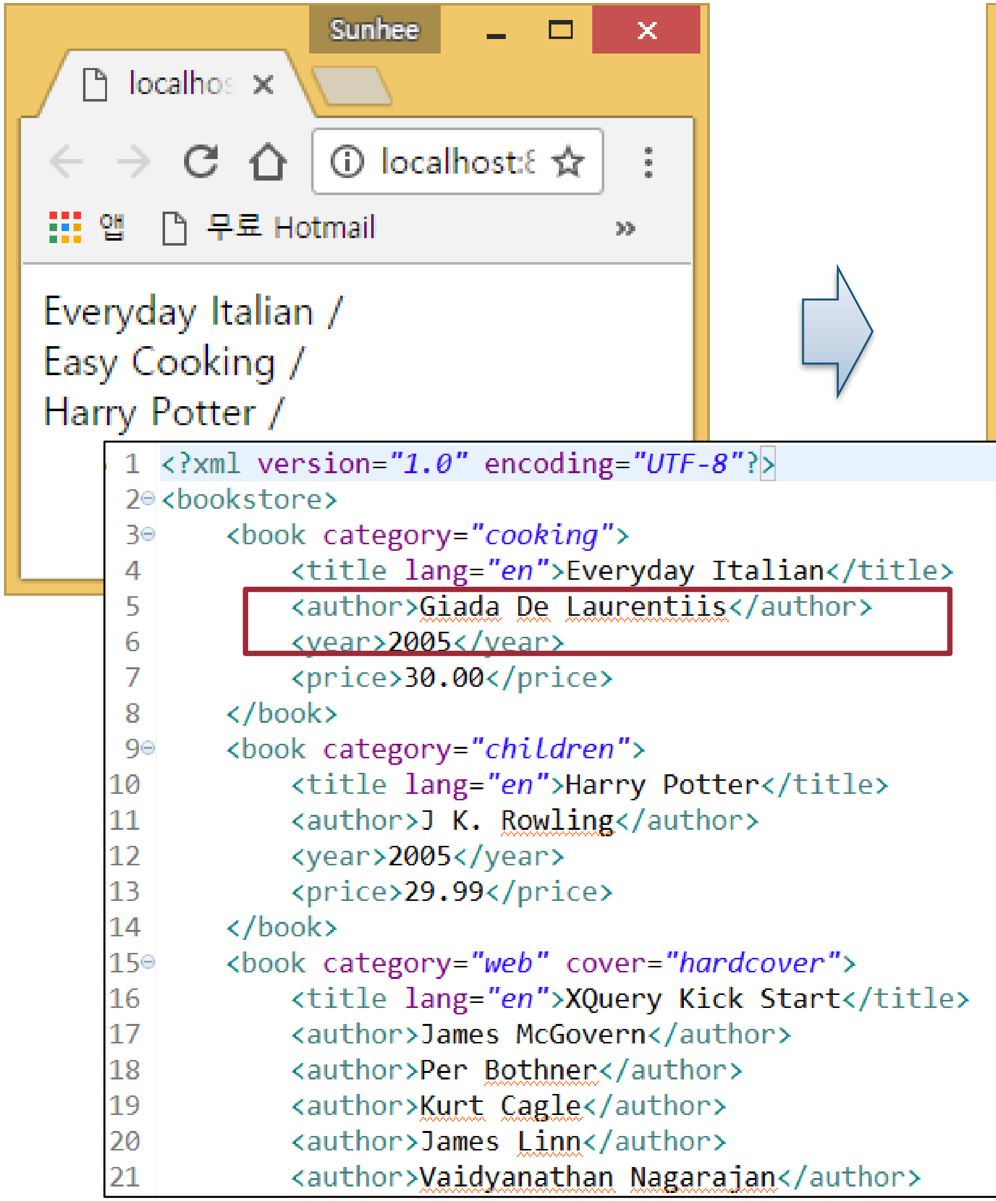
paperback

advance_get_attribute2.html

getAttribute

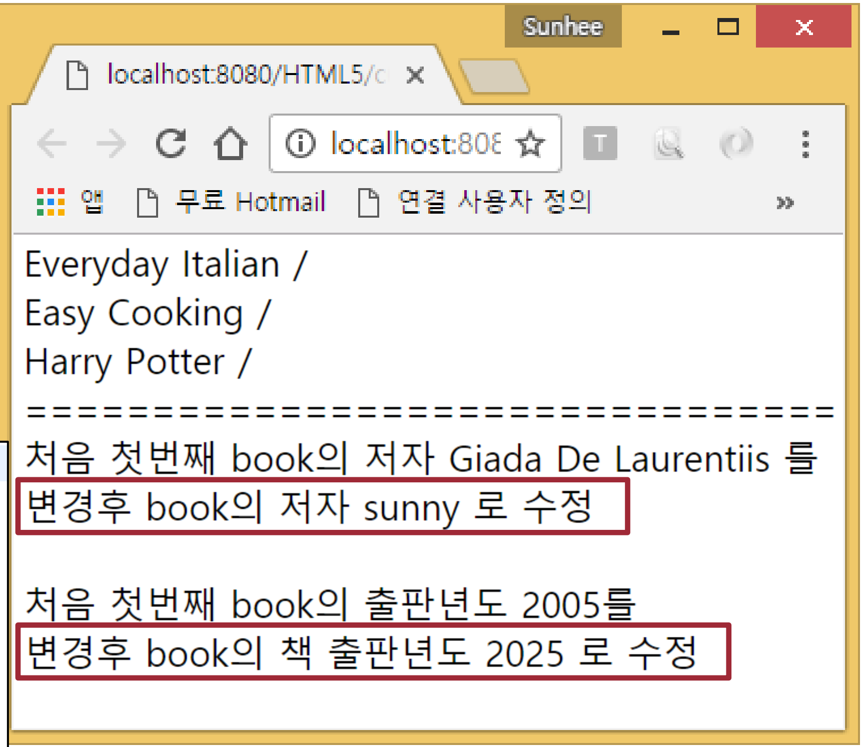
http://localhost:8000/c6_ex2(xml)/3.books/advance_get_attribute2.html

<실습3-3> books – advance_change_element3.html



Everyday Italian /
Easy Cooking /
Harry Potter /

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web" cover="hardcover">
        <title lang="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
```



Everyday Italian /
Easy Cooking /
Harry Potter /
====================================
처음 첫번째 book의 저자 Giada De Laurentiis 를
변경후 book의 저자 sunny 로 수정

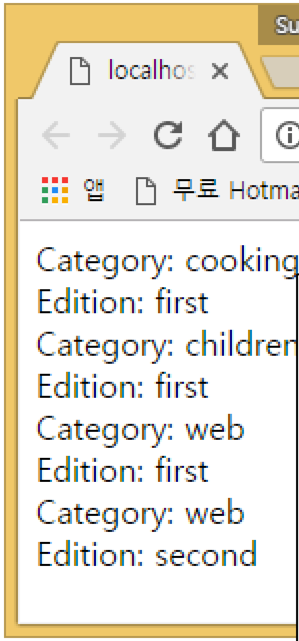
처음 첫번째 book의 출판년도 2005를
변경후 book의 책 출판년도 2025 로 수정

advance_change_element3.html

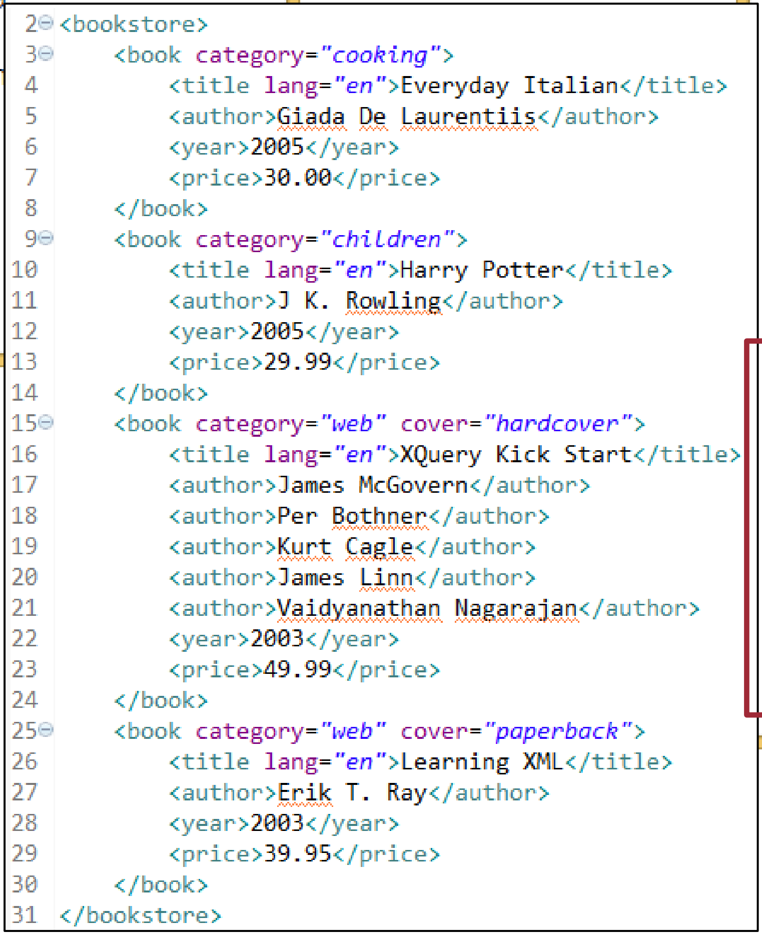
```
x.nodeValue = "sunny";
txt = x.nodeValue;
```

http://localhost:8000/c6_ex2(xml)/3.books/advance_change_element3.html

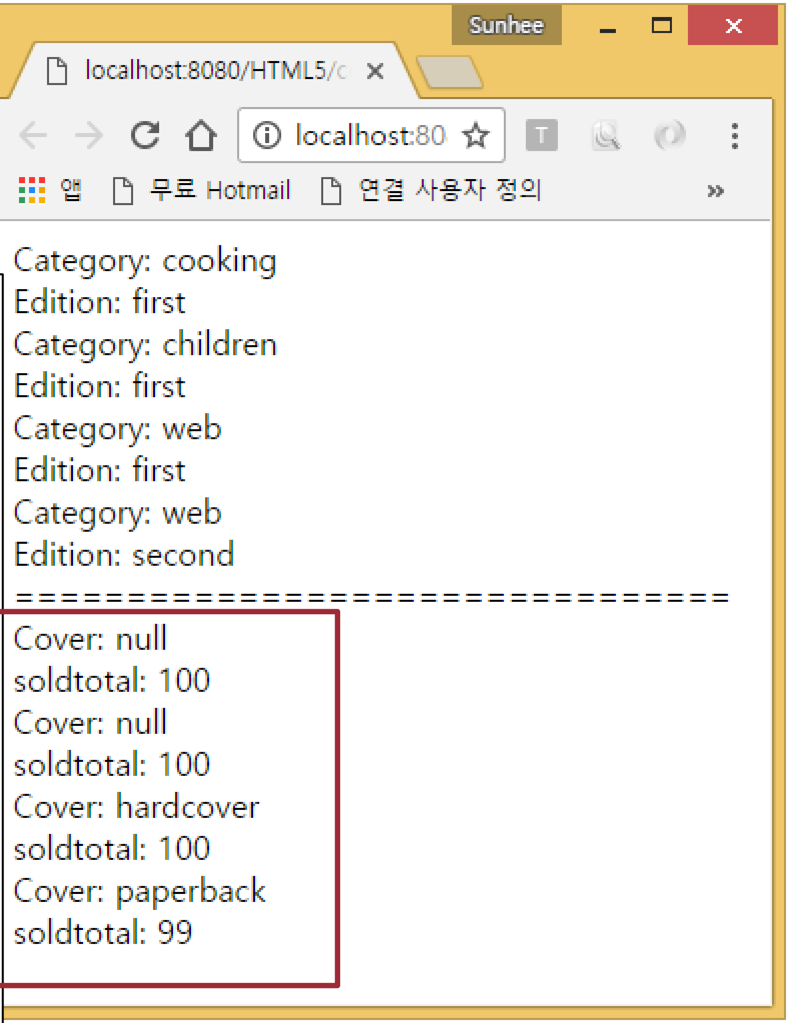
<실습3-4> books – advance_create_attribute4.html



```
Category: cooking
Edition: first
Category: children
Edition: first
Category: web
Edition: first
Category: web
Edition: second
```



```xml
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web" cover="hardcover">
        <title lang="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>2003</year>
        <price>49.99</price>
    </book>
    <book category="web" cover="paperback">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

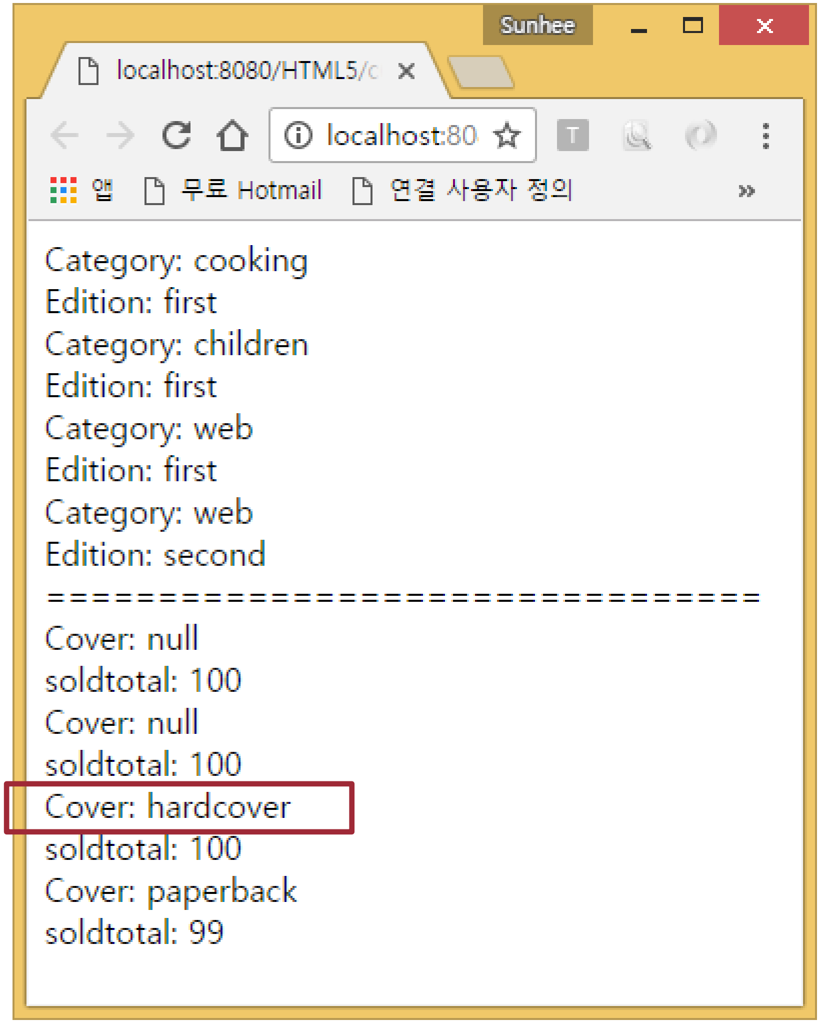


```
Category: cooking
Edition: first
Category: children
Edition: first
Category: web
Edition: first
Category: web
Edition: second
================================
Cover: null
soldtotal: 100
Cover: null
soldtotal: 100
Cover: hardcover
soldtotal: 100
Cover: paperback
soldtotal: 99
```

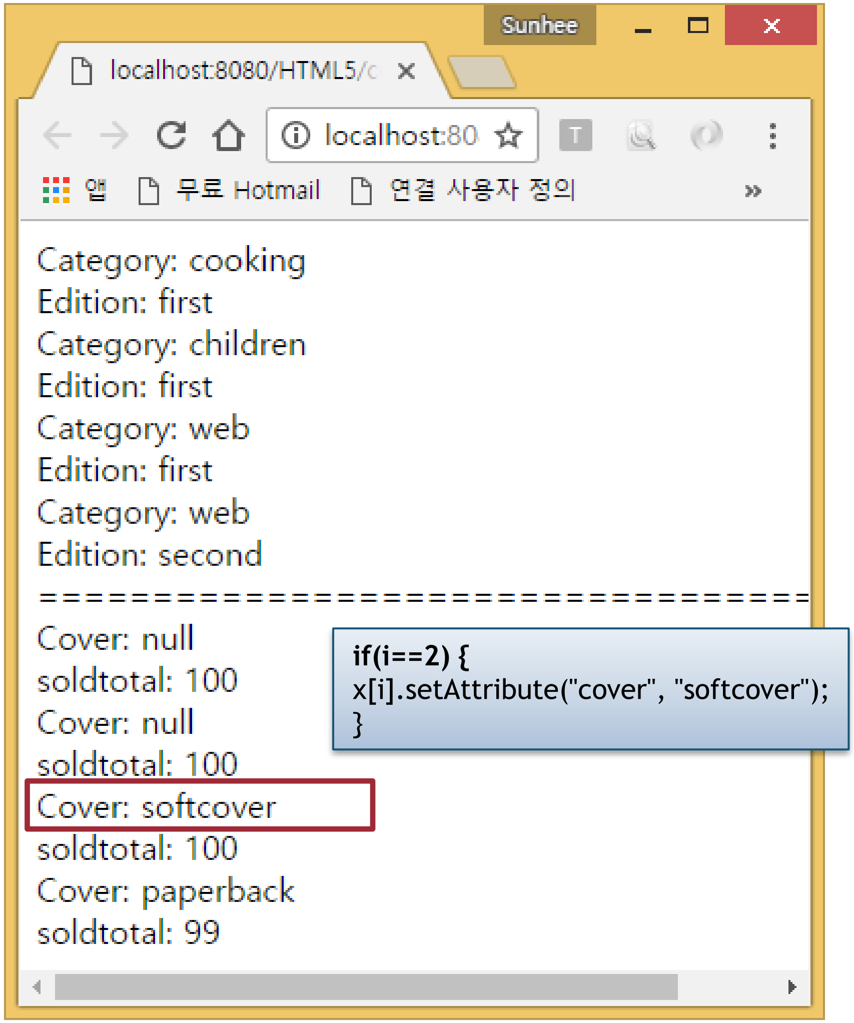
advance_create_attribute4.html

setAttribute

<실습3-4-1> books – advance_create_attribute4_1.html



```
Category: cooking
Edition: first
Category: children
Edition: first
Category: web
Edition: first
Category: web
Edition: second
==================================
Cover: null
soldtotal: 100
Cover: null
soldtotal: 100
Cover: hardcover
soldtotal: 100
Cover: paperback
soldtotal: 99
```

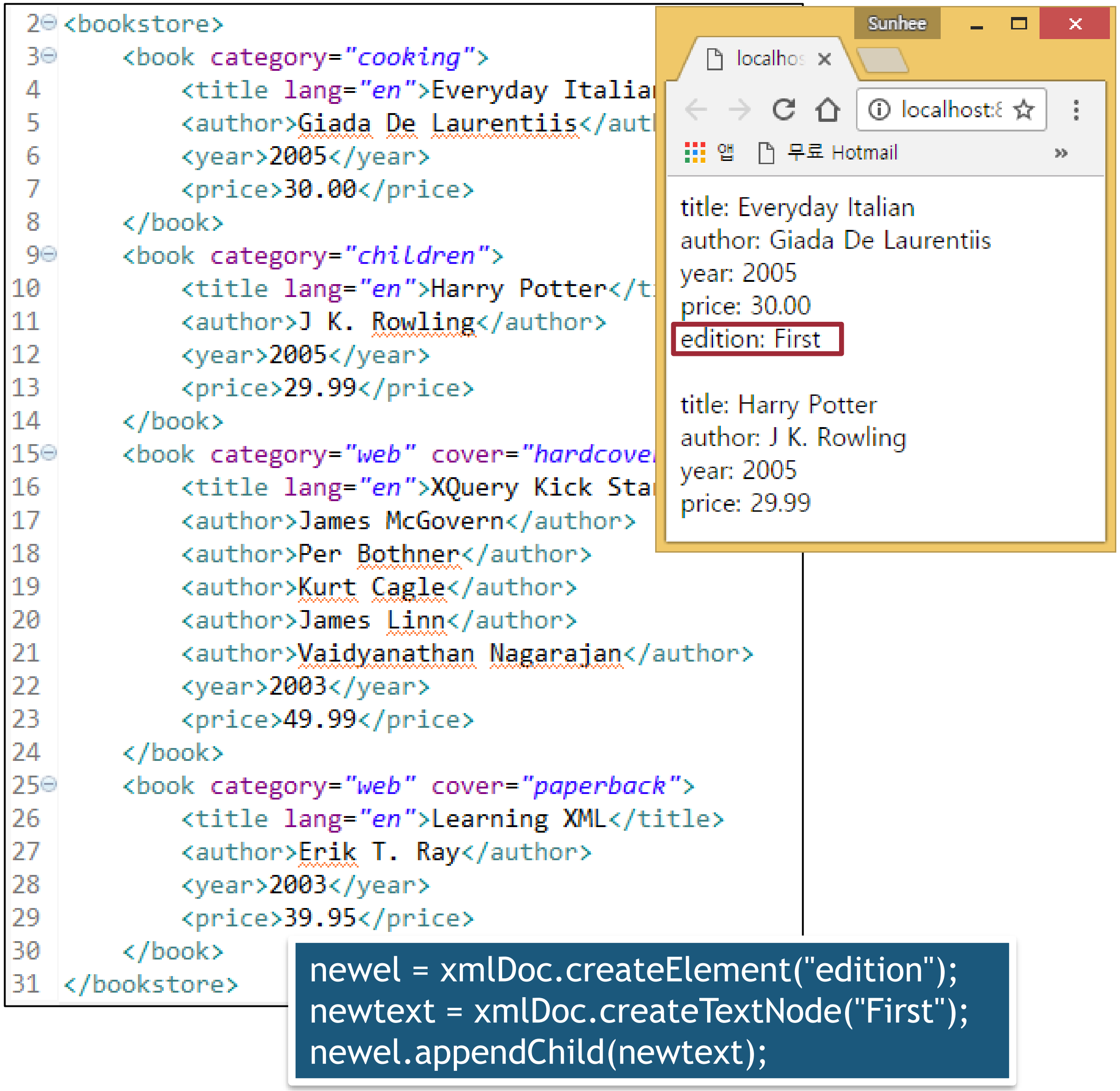


```
Category: cooking
Edition: first
Category: children
Edition: first
Category: web
Edition: first
Category: web
Edition: second
=====================================
Cover: null
soldtotal: 100
Cover: null
soldtotal: 100
Cover: softcover
soldtotal: 100
Cover: paperback
soldtotal: 99
```

```
if(i==2) {
x[i].setAttribute("cover", "softcover");
}
```

setAttribute

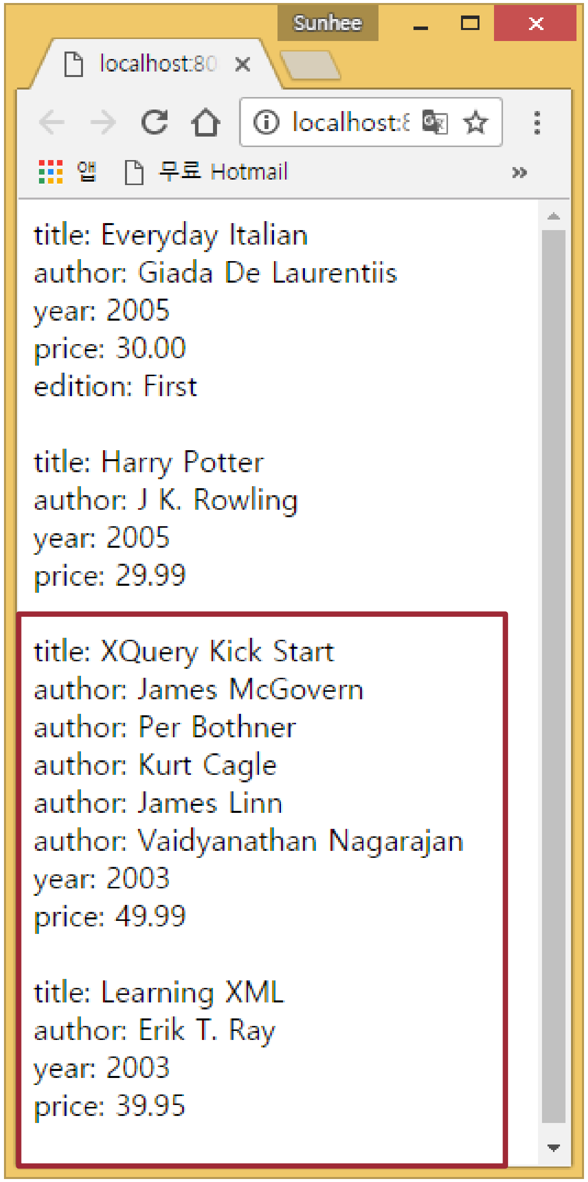
<실습3-5> books – advance_create_element5.html



```
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italia
        <author>Giada De Laurentiis</aut
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</t
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web" cover="hardcove
        <title lang="en">XQuery Kick Sta
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>2003</year>
        <price>49.99</price>
    </book>
    <book category="web" cover="paperback">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

```
title: Everyday Italian
author: Giada De Laurentiis
year: 2005
price: 30.00
edition: First

title: Harry Potter
author: J K. Rowling
year: 2005
price: 29.99
```

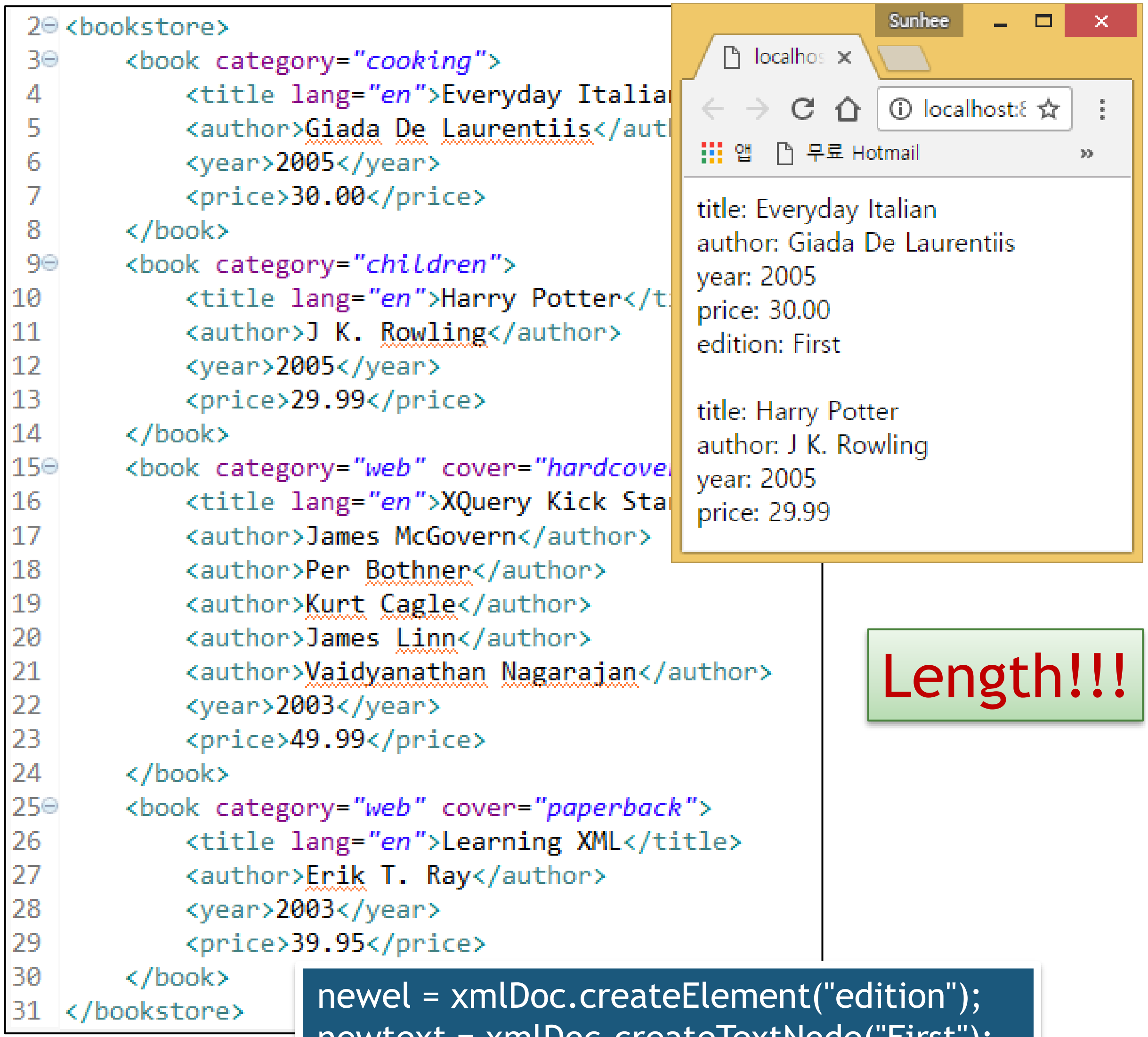
```
newel = xmlDoc.createElement("edition");
newtext = xmlDoc.createTextNode("First");
newel.appendChild(newtext);
```



```
title: Everyday Italian
author: Giada De Laurentiis
year: 2005
price: 30.00
edition: First

title: Harry Potter
author: J K. Rowling
year: 2005
price: 29.99

title: XQuery Kick Start
author: James McGovern
author: Per Bothner
author: Kurt Cagle
author: James Linn
author: Vaidyanathan Nagarajan
year: 2003
price: 49.99

title: Learning XML
author: Erik T. Ray
year: 2003
price: 39.95
```

advance_create_element5.html

<실습3-5-1> books – advance_create_element5_1.html

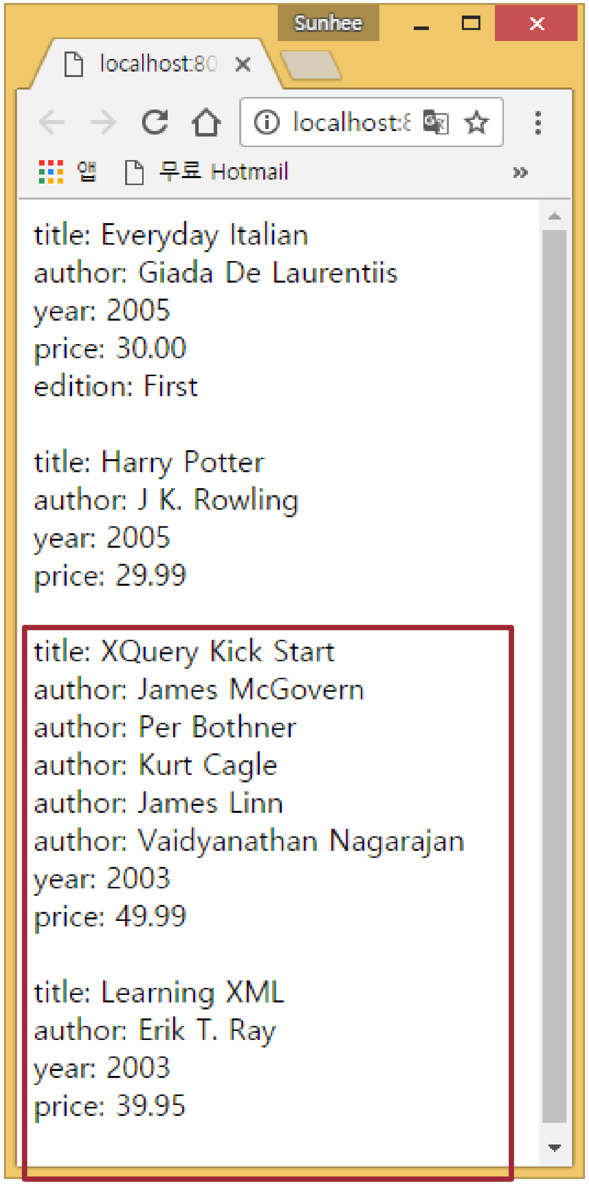


```
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italia
        <author>Giada De Laurentiis</aut
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</t
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web" cover="hardcove
        <title lang="en">XQuery Kick Sta
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>2003</year>
        <price>49.99</price>
    </book>
    <book category="web" cover="paperback">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

```
title: Everyday Italian
author: Giada De Laurentiis
year: 2005
price: 30.00
edition: First

title: Harry Potter
author: J K. Rowling
year: 2005
price: 29.99
```

Length!!!

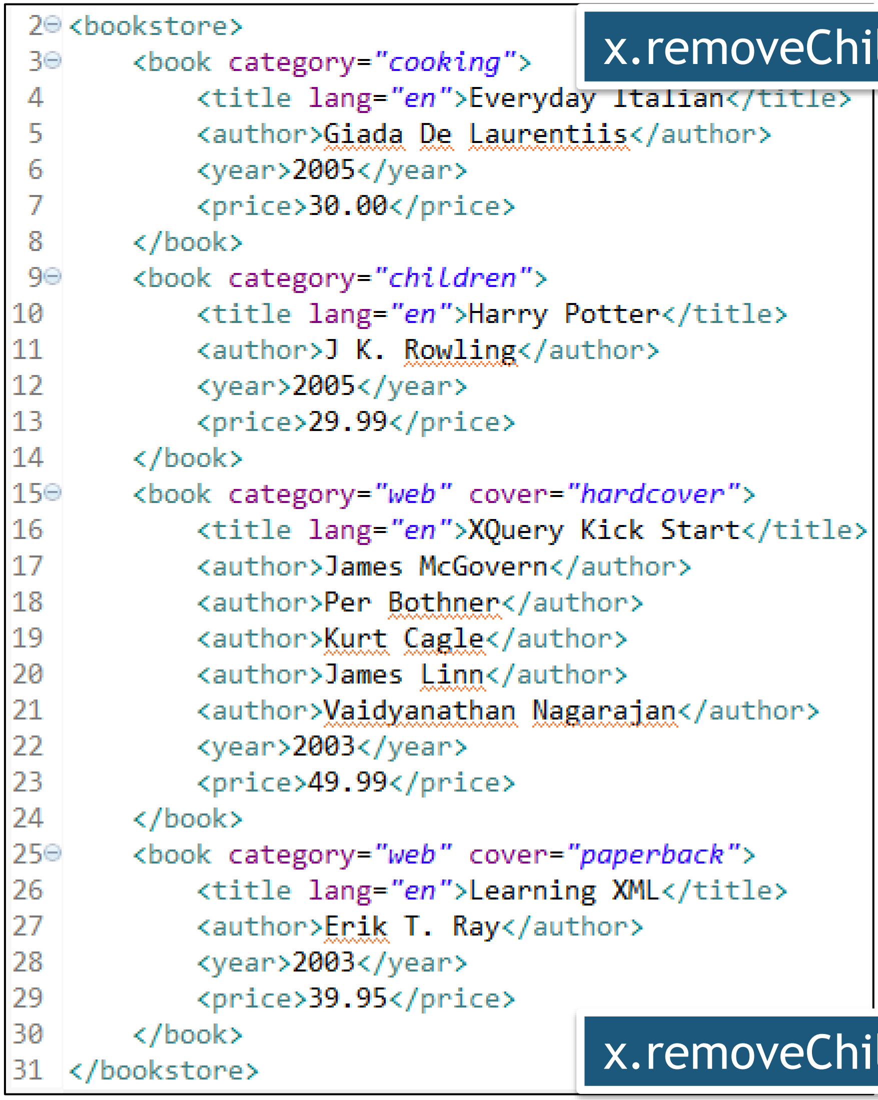
```
newel = xmlDoc.createElement("edition");
newtext = xmlDoc.createTextNode("First");
newel.appendChild(newtext);
```



```
title: Everyday Italian
author: Giada De Laurentiis
year: 2005
price: 30.00
edition: First

title: Harry Potter
author: J K. Rowling
year: 2005
price: 29.99

title: XQuery Kick Start
author: James McGovern
author: Per Bothner
author: Kurt Cagle
author: James Linn
author: Vaidyanathan Nagarajan
year: 2003
price: 49.99

title: Learning XML
author: Erik T. Ray
year: 2003
price: 39.95
```

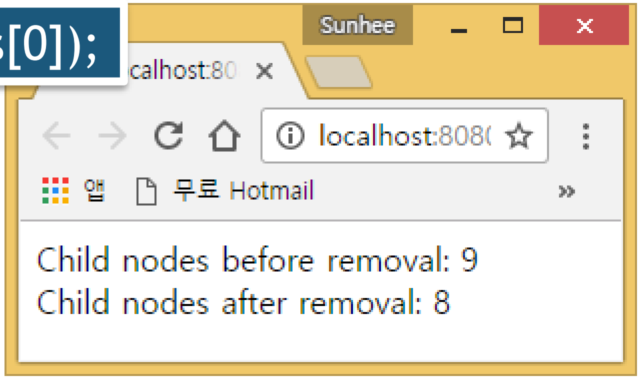
advance_create_element5_1.html

<실습3-6> books – advance_remove_element6.html



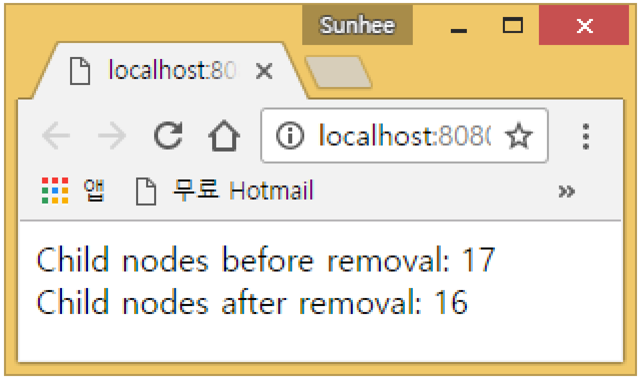
```
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web" cover="hardcover">
        <title lang="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>2003</year>
        <price>49.99</price>
    </book>
    <book category="web" cover="paperback">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

x.removeChild(x.childNodes[0]);



```
Child nodes before removal: 9
Child nodes after removal: 8
```

removeChild!!!

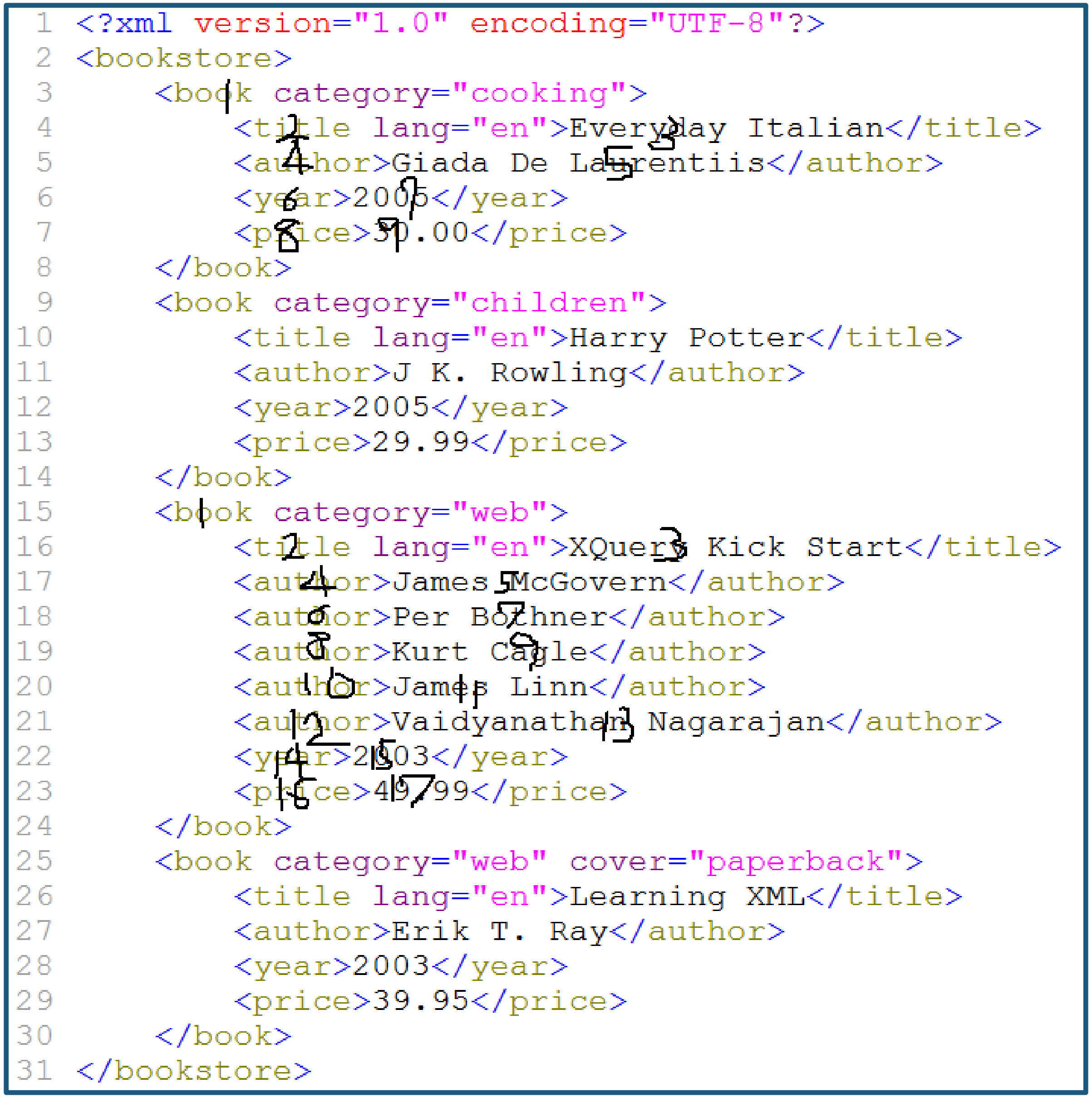


```
Child nodes before removal: 17
Child nodes after removal: 16
```

advance_remove_element6.html

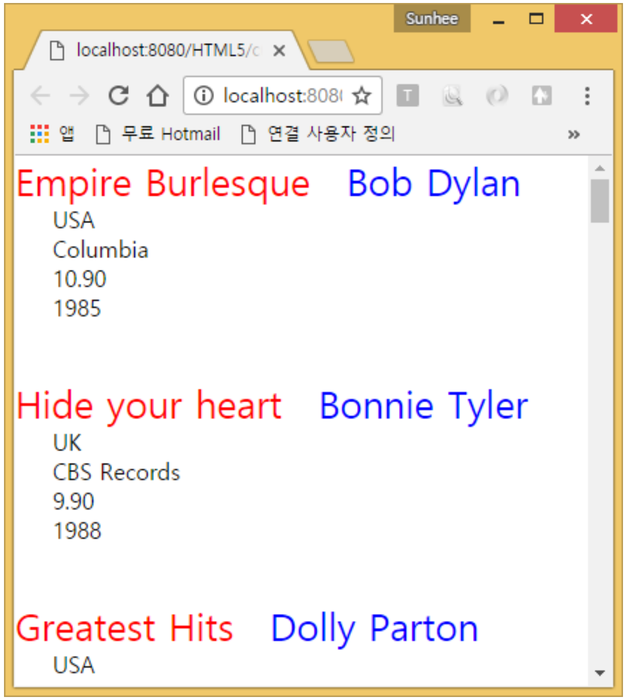
x.removeChild(x.childNodes[2]);

<실습3-6-1> books - advance_remove_element6.html 구조변경 확인



```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
    <book category="cooking">
        <title lang="en">Everyday Italian</title>
        <author>Giada De Laurentiis</author>
        <year>2005</year>
        <price>30.00</price>
    </book>
    <book category="children">
        <title lang="en">Harry Potter</title>
        <author>J K. Rowling</author>
        <year>2005</year>
        <price>29.99</price>
    </book>
    <book category="web">
        <title lang="en">XQuery Kick Start</title>
        <author>James McGovern</author>
        <author>Per Bothner</author>
        <author>Kurt Cagle</author>
        <author>James Linn</author>
        <author>Vaidyanathan Nagarajan</author>
        <year>2003</year>
        <price>49.99</price>
    </book>
    <book category="web" cover="paperback">
        <title lang="en">Learning XML</title>
        <author>Erik T. Ray</author>
        <year>2003</year>
        <price>39.95</price>
    </book>
</bookstore>
```

<실습4-1> cd_catalog – cd_catalog.xml

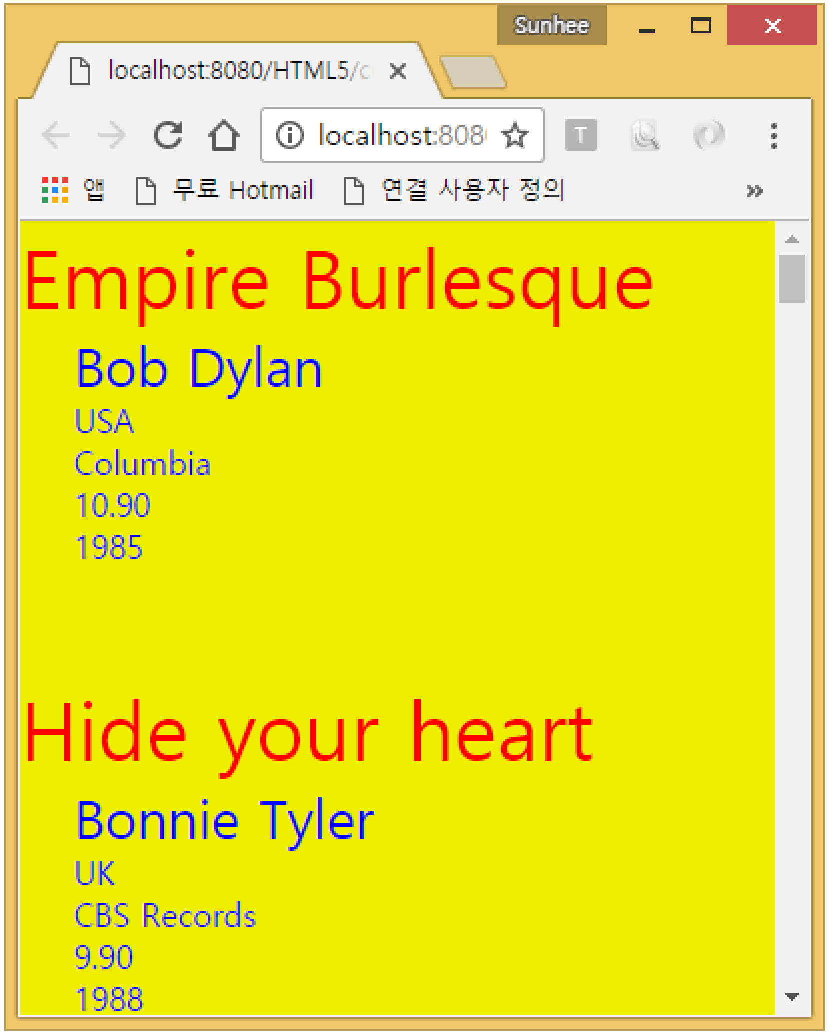


Empire Burlesque Bob Dylan
USA
Columbia
10.90
1985

Hide your heart Bonnie Tyler
UK
CBS Records
9.90
1988

Greatest Hits Dolly Parton
USA

cd_catalog.css
만들기

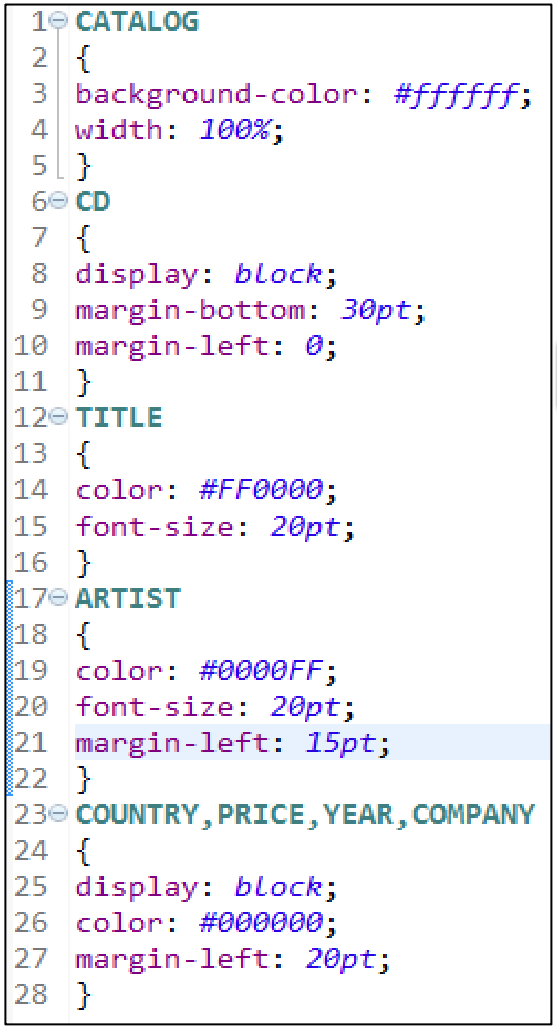


Empire Burlesque
Bob Dylan
USA
Columbia
10.90
1985

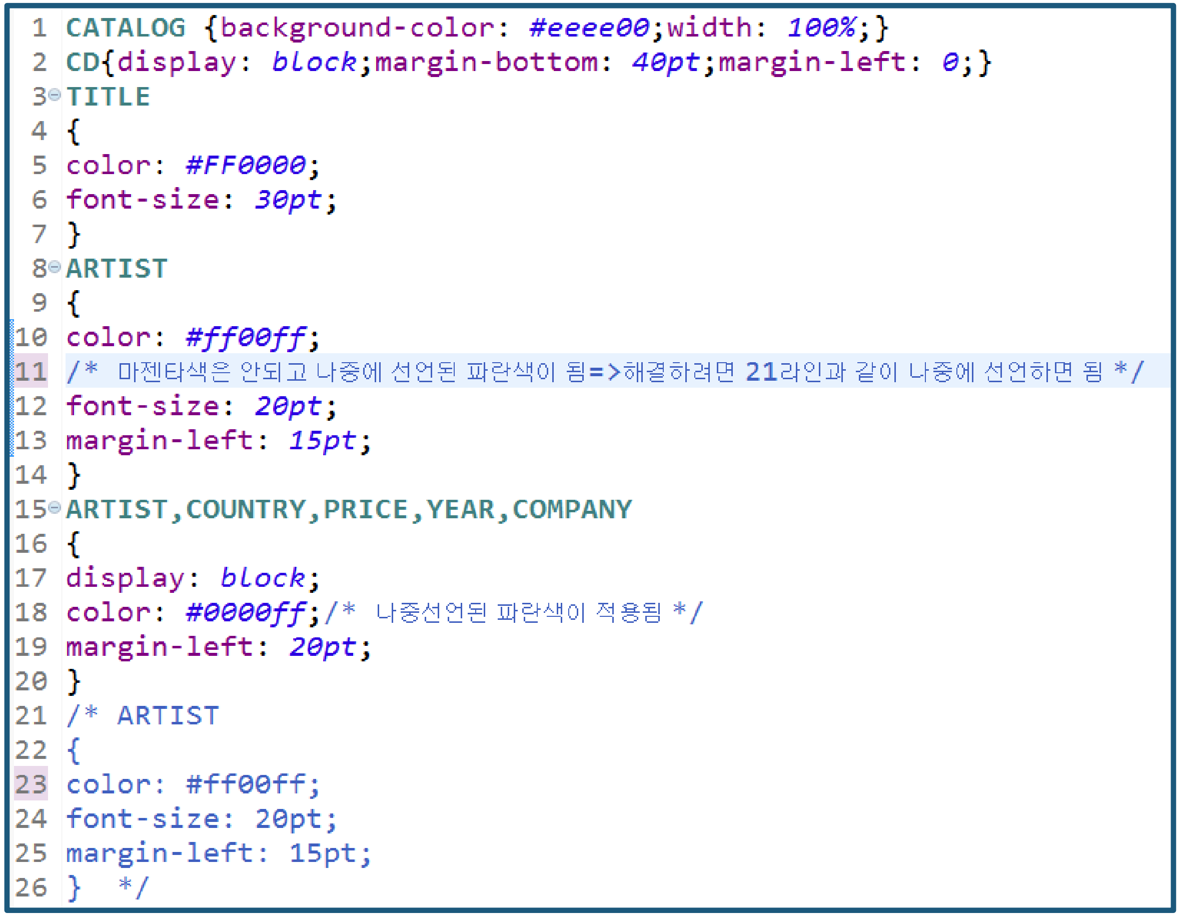
Hide your heart
Bonnie Tyler
UK
CBS Records
9.90
1988

cd_catalog.xml

<실습4-1> cd_catalog - cd_catalog.css 만들기



```
CATALOG
{
background-color: #ffffff;
width: 100%;
}
CD
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
TITLE
{
color: #FF0000;
font-size: 20pt;
}
ARTIST
{
color: #0000FF;
font-size: 20pt;
margin-left: 15pt;
}
COUNTRY,PRICE,YEAR,COMPANY
{
display: block;
color: #000000;
margin-left: 20pt;
}
```



```
 1 CATALOG {background-color: #eeee00;width: 100%;}
 2 CD{display: block;margin-bottom: 40pt;margin-left: 0;}
 3 TITLE
 4 {
 5 color: #FF0000;
 6 font-size: 30pt;
 7 }
 8 ARTIST
 9 {
10 color: #ff00ff;
11 /* 마젠타색은 안되고 나중에 선언된 파란색이 됨=>해결하려면 21라인과 같이 나중에 선언하면 됨 */
12 font-size: 20pt;
13 margin-left: 15pt;
14 }
15 ARTIST,COUNTRY,PRICE,YEAR,COMPANY
16 {
17 display: block;
18 color: #0000ff;/* 나중선언된 파란색이 적용됨 */
19 margin-left: 20pt;
20 }
21 /* ARTIST
22 {
23 color: #ff00ff;
24 font-size: 20pt;
25 margin-left: 15pt;
26 }  */
```

cd_catalog.css

〈실습4-2〉 cd_catalog – xmlapp_cd_catalog2.html

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
    <CD>
        <TITLE>Empire Burlesque</TITLE>
        <ARTIST>Bob Dylan</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>Columbia</COMPANY>
        <PRICE>10.90</PRICE>
        <YEAR>1985</YEAR>
    </CD>
    <CD>
        <TITLE>Hide your heart</TITLE>
        <ARTIST>Bonnie Tyler</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>CBS Records</COMPANY>
        <PRICE>9.90</PRICE>
        <YEAR>1988</YEAR>
    </CD>
    <CD>
        <TITLE>Greatest Hits</TITLE>
```

......

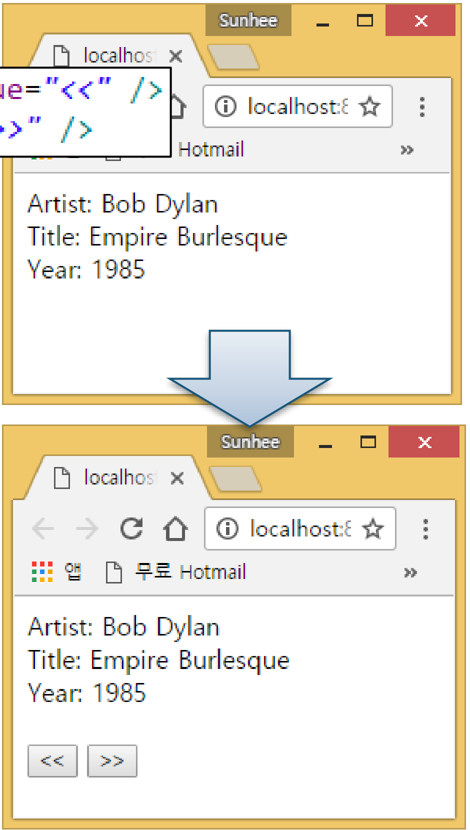
```xml
    <CD>
        <TITLE>Red</TITLE>
        <ARTIST>The Communards</ARTIST>
        <COUNTRY>UK</COUNTRY>
        <COMPANY>London</COMPANY>
        <PRICE>7.80</PRICE>
        <YEAR>1987</YEAR>
    </CD>
    <CD>
        <TITLE>Unchain my heart</TITLE>
        <ARTIST>Joe Cocker</ARTIST>
        <COUNTRY>USA</COUNTRY>
        <COMPANY>EMI</COMPANY>
        <PRICE>8.20</PRICE>
        <YEAR>1987</YEAR>
    </CD>
</CATALOG>
```

cd_catalog.xml

```html
<input type="button" onclick="previous()" value="<<" />
<input type="button" onclick="next()" value=">>" />
```

```javascript
function next() {
    if (i < x.length - 1) {
        i++;
        displayCD();
    }
}

function previous() {
    if (i > 0) {
        i--;
        displayCD();
    }
}
```

각 <CD> 로 좌우로 이동!



Artist: Bob Dylan
Title: Empire Burlesque
Year: 1985

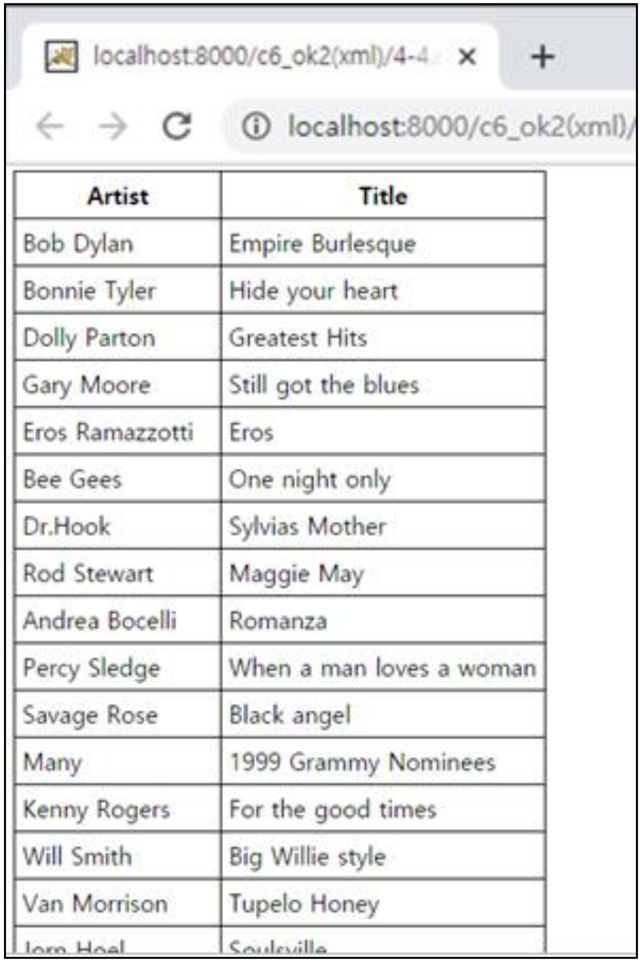
Artist: Bob Dylan
Title: Empire Burlesque
Year: 1985

xmlapp_cd_catalog2.html

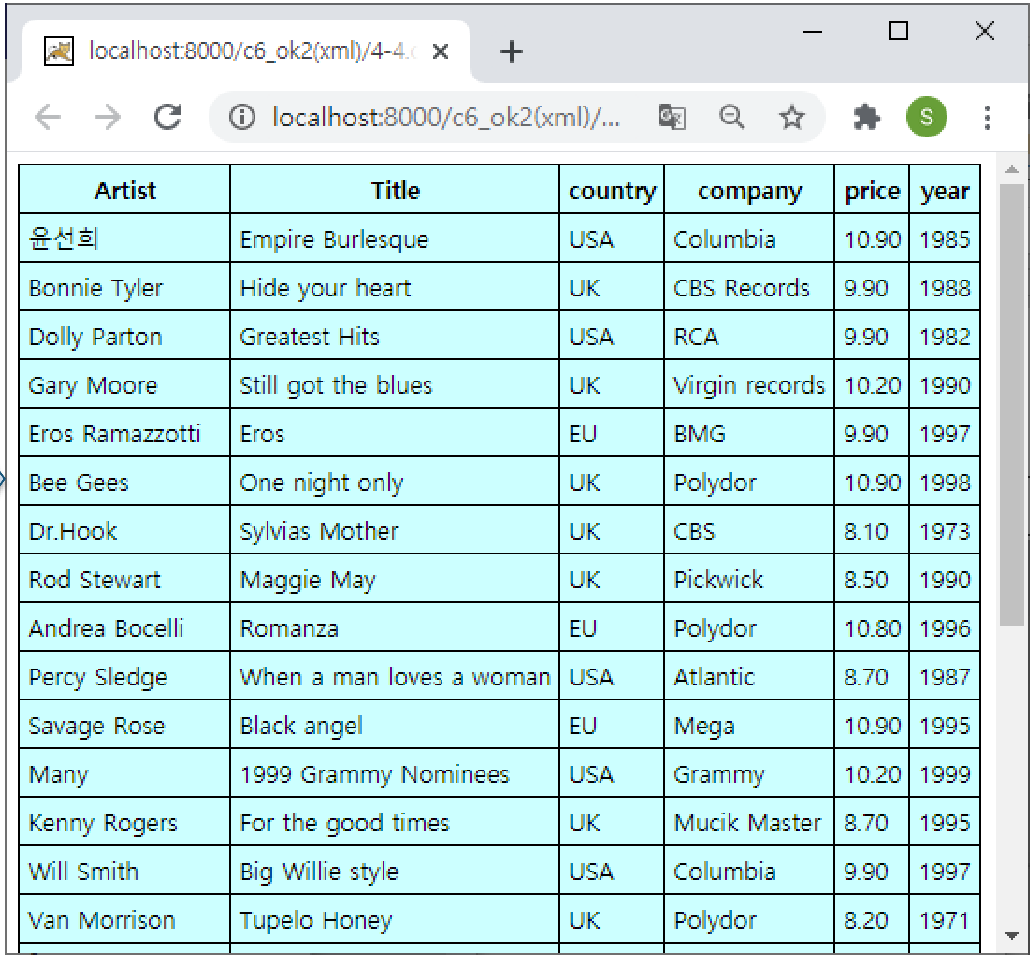
```javascript
function displayCD() {
    artist = (x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue);
    title = (x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue);
    year = (x[i].getElementsByTagName("YEAR")[0].childNodes[0].nodeValue);
    txt = "Artist: " + artist + "<br>Title: " + title + "<br>Year: " + year;
    document.getElementById("showCD").innerHTML = txt;
}
```

반복되는 부분을 함수로 만들어 호출하여 사용!

<실습4-3> cd_catalog – xmlhtml_cd_catalog3.html



Artist	Title
Bob Dylan	Empire Burlesque
Bonnie Tyler	Hide your heart
Dolly Parton	Greatest Hits
Gary Moore	Still got the blues
Eros Ramazzotti	Eros
Bee Gees	One night only
Dr.Hook	Sylvias Mother
Rod Stewart	Maggie May
Andrea Bocelli	Romanza
Percy Sledge	When a man loves a woman
Savage Rose	Black angel
Many	1999 Grammy Nominees
Kenny Rogers	For the good times
Will Smith	Big Willie style
Van Morrison	Tupelo Honey
Jorn Hoel	Soulsville


localhost:8000/c6_ok2(xml)/4-4.
localhost:8000/c6_ok2(xml)/...
S

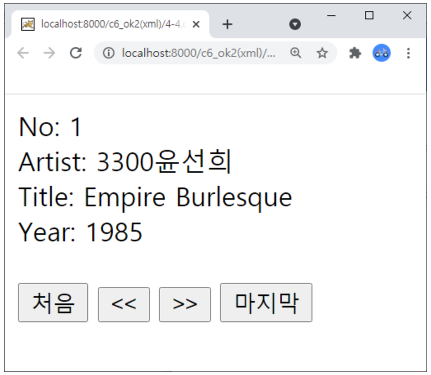
Artist	Title	country	company	price	year
윤선희	Empire Burlesque	USA	Columbia	10.90	1985
Bonnie Tyler	Hide your heart	UK	CBS Records	9.90	1988
Dolly Parton	Greatest Hits	USA	RCA	9.90	1982
Gary Moore	Still got the blues	UK	Virgin records	10.20	1990
Eros Ramazzotti	Eros	EU	BMG	9.90	1997
Bee Gees	One night only	UK	Polydor	10.90	1998
Dr.Hook	Sylvias Mother	UK	CBS	8.10	1973
Rod Stewart	Maggie May	UK	Pickwick	8.50	1990
Andrea Bocelli	Romanza	EU	Polydor	10.80	1996
Percy Sledge	When a man loves a woman	USA	Atlantic	8.70	1987
Savage Rose	Black angel	EU	Mega	10.90	1995
Many	1999 Grammy Nominees	USA	Grammy	10.20	1999
Kenny Rogers	For the good times	UK	Mucik Master	8.70	1995
Will Smith	Big Willie style	USA	Columbia	9.90	1997
Van Morrison	Tupelo Honey	UK	Polydor	8.20	1971

xmlhtml_cd_catalog3.html

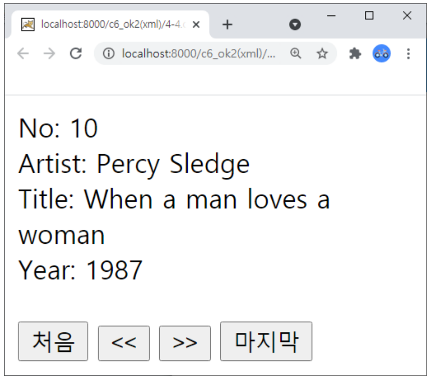
http://localhost:8000/c6_ex2(xml)/4.cd_catalog/xmlhtml_cd_catalog3.html

<실습> cd_catalog – xmlapp_cd_catalog2_1.html

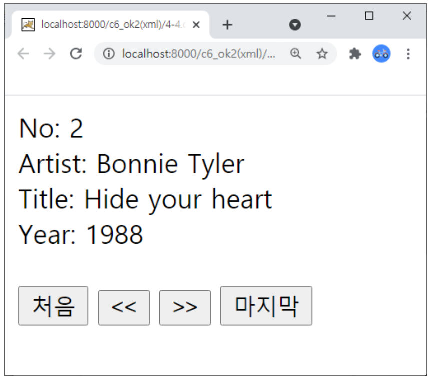
<CD> 의 처음과 끝으로 이동!


localhost:8000/c6_ok2(xml)/4-4
localhost:8000/c6_ok2(xml)/...
처음
<<
>>
마지막

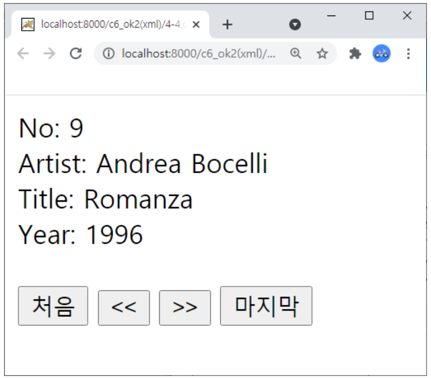
No: 1
Artist: 3300윤선희
Title: Empire Burlesque
Year: 1985


localhost:8000/c6_ok2(xml)/4-4
localhost:8000/c6_ok2(xml)/...
처음
<<
>>
마지막

No: 10
Artist: Percy Sledge
Title: When a man loves a woman
Year: 1987


localhost:8000/c6_ok2(xml)/4-4
localhost:8000/c6_ok2(xml)/...
처음
<<
>>
마지막

No: 2
Artist: Bonnie Tyler
Title: Hide your heart
Year: 1988


localhost:8000/c6_ok2(xml)/4-4
localhost:8000/c6_ok2(xml)/...
처음
<<
>>
마지막

No: 9
Artist: Andrea Bocelli
Title: Romanza
Year: 1996

```
function first(){


}

function end(){


}
```

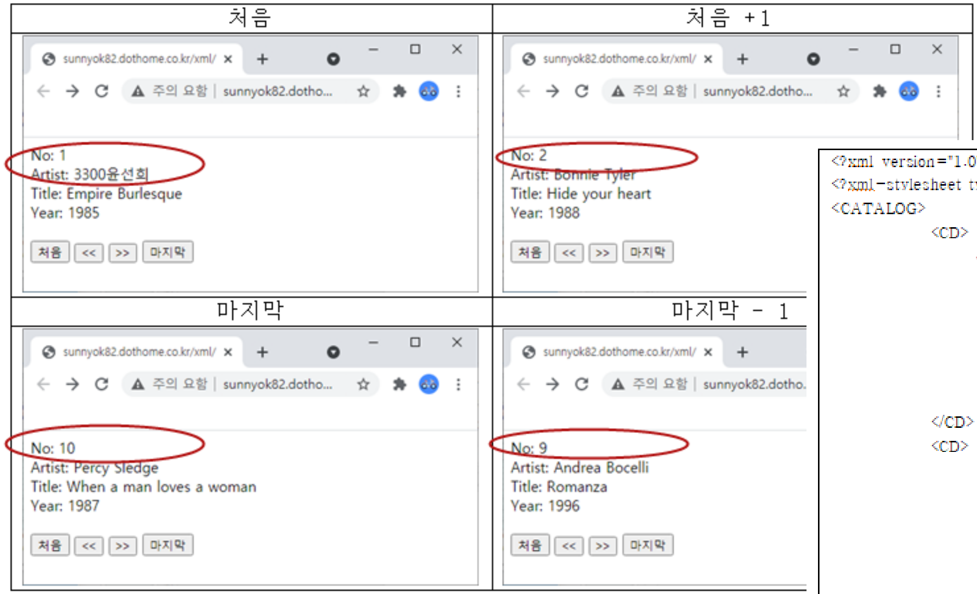
반복되는 부분을 함수로 만들어 호출하여 사용!

http://localhost:8000/c6_ex2(xml)/4.cd_catalog/xmlapp_cd_catalog2_1.html

xmlapp_cd_catalog2_1.html 로 완성후 index.html로 수정하여 dothome 올리기

<실습> XML 구조 보고서 제출 형식 설명

실행화면



처음

No: 1
Artist: 3300윤선희
Title: Empire Burlesque
Year: 1985

처음 +1

No: 2
Artist: Bonnie Tyler
Title: Hide your heart
Year: 1988

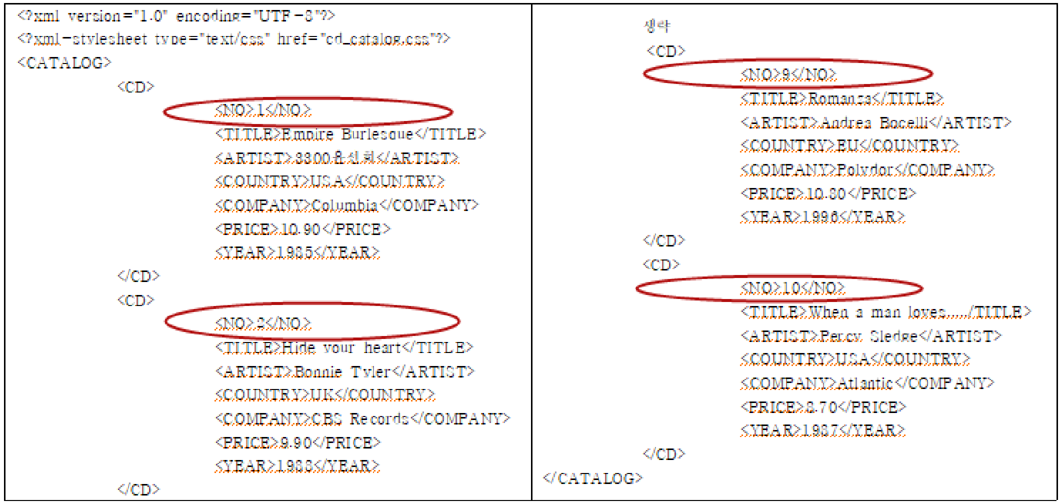
마지막

No: 10
Artist: Percy Sledge
Title: When a man loves a woman
Year: 1987

마지막 - 1

No: 9
Artist: Andrea Bocelli
Title: Romanza
Year: 1996

xml 구조



```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/css" href="cd_catalog.css"?>
<CATALOG>
        <CD>
                <NO>1</NO>
                <TITLE>Empire Burlesque</TITLE>
                <ARTIST>3300윤선희</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>Columbia</COMPANY>
                <PRICE>10.90</PRICE>
                <YEAR>1985</YEAR>
        </CD>
        <CD>
                <NO>2</NO>
                <TITLE>Hide your heart</TITLE>
                <ARTIST>Bonnie Tyler</ARTIST>
                <COUNTRY>UK</COUNTRY>
                <COMPANY>CBS Records</COMPANY>
                <PRICE>9.90</PRICE>
                <YEAR>1988</YEAR>
        </CD>
```

```
        생략
        <CD>
                <NO>9</NO>
                <TITLE>Romanza</TITLE>
                <ARTIST>Andrea Bocelli</ARTIST>
                <COUNTRY>EU</COUNTRY>
                <COMPANY>Polydor</COMPANY>
                <PRICE>10.80</PRICE>
                <YEAR>1996</YEAR>
        </CD>
        <CD>
                <NO>10</NO>
                <TITLE>When a man loves...../TITLE>
                <ARTIST>Percy Sledge</ARTIST>
                <COUNTRY>USA</COUNTRY>
                <COMPANY>Atlantic</COMPANY>
                <PRICE>8.70</PRICE>
                <YEAR>1987</YEAR>
        </CD>
</CATALOG>
```

※ 주의사항
- 실행화면에 학번과 이름이 나타나도록 하고 데이터를 변형하여 제출
- 모든 실행 화일에 하이퍼링크가 반드시 있어야 함
- cd_catalog.xml의 내용을 모두 다르게 작성하고 <CD>를 10개 이상 넣기
- 맨앞의 node에 번호 요소를 넣어야 함 (1 ~ 10)
- 프로그램 – html, css, xml 등등

```
<CD>
   <No>1</No>
   <TITLE>Empire Burlesque</TITLE>
   <ARTIST>3300윤선희</ARTIST>
   <COUNTRY>USA</COUNTRY>
   <COMPANY>Columbia</COMPANY>
   <PRICE>10.90</PRICE>
   <YEAR>1985</YEAR>
</CD>
```

Q & A