

---

<JSTORM>

---

JAVA XML



JSTORM  
<http://www.jstorm.pe.kr>

---

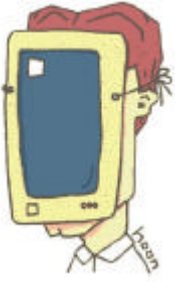
### Document Information

Document title:	JAVA XML
Document file name:	JAVA XML_ .doc
Revision number:	<1.0>
Issued by:	< > jihyun@nownuri.net
Issue Date:	<2000/8/01 >
Status:	Draft

### Content Information

Audience	JAVA XML
	,
Abstract	o XML
	- XML, DTD
	o JAVA DOM
	o JSP XML
Reference	
Benchmark information	

### Document Approvals

	Signature	date
		
	Signature	date

### Revision History

<u>Revision</u>	<u>Date</u>	<u>Author</u>	<u>Description of change</u>

## Table of Contents

<b>XML (Extensible Markup Language) ? .....</b>	<b>6</b>
<b>XML .....</b>	<b>6</b>
<b>DTD (Document Type Definition).....</b>	<b>7</b>
<b>    Parsing.....</b>	<b>8</b>
<b>SAX &amp; DOM.....</b>	<b>11</b>
<b>XML JSP .....</b>	<b>12</b>
<b>    XML .....</b>	<b>15</b>
<b>.....</b>	<b>16</b>
<b>.....</b>	<b>16</b>

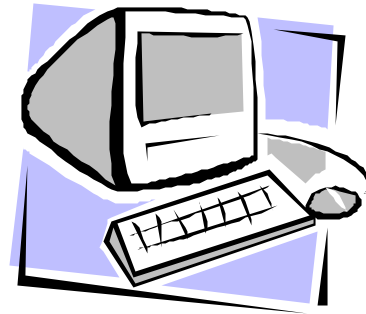


---

# JAVA XML

XML  
XML  
Sun Microsystem jon Bosak XML JAVA가  
XML

가 가



## XML (Extensible Markup Language) ?

SGML

가  
가 hello

HTML

가

<h1> < font = bold> Hello!! </font></h1>

XML

< greeting> Hello !! < /greeting>

가?

HTML

Hello!

XML

Hello!

가

?

??

가Data

Data

## XML

1)

greeting>

Hello

<

.(

DTD가

)

HTML

2)

XML data

```
<book>
  <title> Java Bible </title>
  <publish> yung jin </publish>
  <year> 1999</year>
  <price>30000 </price>
</book>
```

XML

가

```
. book title, publish, year, price 가 .
```

3)

XML

DTD(Document Type Definition)

가

## DTD (Document Type Definition)

DTD

```
<?xml version="1.0"?>

<!DOCTYPE book [
  <!ELEMENT book ( title, publisher, year, price)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT publish (#PCDATA)>
  <!ELEMENT year (#PCDATA)>
  <!ELEMENT price (#PCDATA)>
]>

<book>
  <title> Java Bible </title>
  <publish> yung jin </publish>
  <year> 1999</year>
  <price>30000 </price>
</book>
```

DTD

XML

DTD

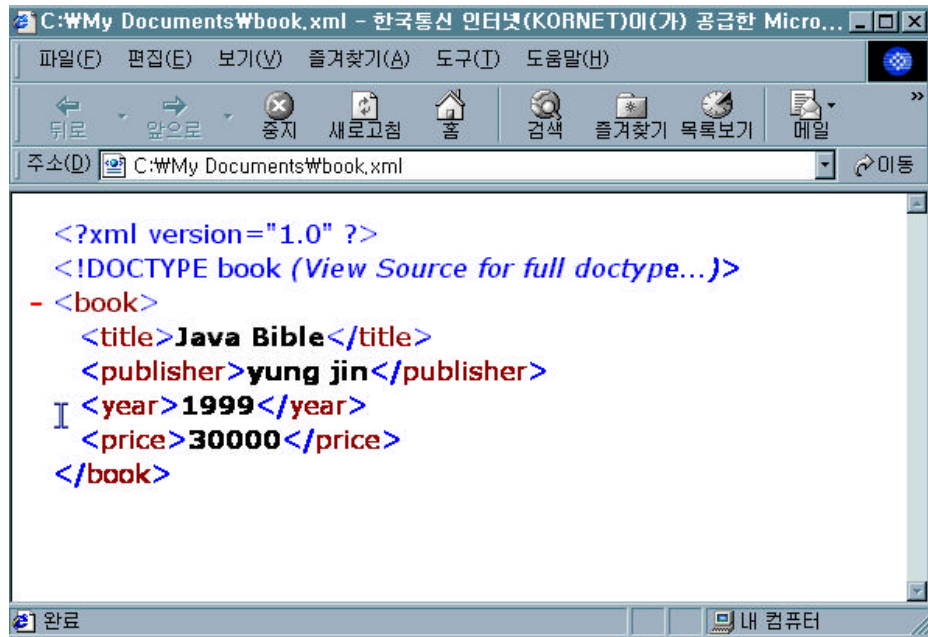
XML

<!DOCTYPE doctype SYSTEM "book.dtd">

DTD

DTD <!DOCTYPE> . XML DTD  
 book title, publisher, year, price  
 XML Document Instance year DTD  
 가  
 <!ELEMENT> title, publisher, year, price  
 가 #PCDATA (Parsed Character Data)  
 ) XML

IE5 ??  
 Internet Explorer 5 XML XML (book.xml)  
 Explorer DTD XML 가  
 가 error message가



### Parsing

(Parser) ??  
 Parser 가 XML



XML 가 XML

### XML

~~IBM XML for 2.0~~ (<http://alphaworks.ibm.com/tech/xml/>)  
~~X Realease 2~~ (<http://java.sun.com/xml/>)  
~~DataChannel XJParser~~ (<http://xdev.datachannel.com/downloads/xjparser/>)  
~~XML for 2~~ (<http://technet.oracle.com/tech/xml/>)

### XML

1.1 JDK

~~<http://developer.java.sun.com/developer/earlyAccess/xml/index.html>~~

### XML parser 1.0

parser.jar jaxp.jar classpath

Working Directory - C:\jdk1.3\

Class Path - C:\Program Files\JavaSoft\Jaxp1.0.1\jaxp.jar;C:\Program Files\JavaSoft\Jaxp1.0.1\parser.jar;C:\PACKAGE;C:\JDK1.3\LIB\TOOLS.JAR;C:\LANG\KAWA3.51\KAWACLASSES.ZIP;C:\SDK2.1\SERVER.JAR;C:\SDK2.1\SERVLET.JAR;C:\JDK1.3\JRE\LIB\RT.JAR;.;c:\jdk1.3\jre\lib\i18n.jar

### SimpleParser

```
import java.io.File;
import com.sun.xml.parser.Resolver;
import com.sun.xml.tree.XmlDocument;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import org.xml.sax.SAXParseException;

public class Simpleparser
{
    public static void main (String argv [])
    {
        InputSource input;
```

```
XmlDocument doc;

if (argv.length != 1)
{
    System.err.println("Usage: java Simpleparser [filename]");
    System.exit (1);
}

try
{
    input = Resolver.createInputSource (new File(argv [0]));
    doc = XmlDocument.createXmlDocument (input, false);
    doc.getDocumentElement ().normalize ();
    doc.write (System.out);
}
catch (SAXParseException err)
{
    System.out.println ("** Parsing error" + ", line " +
        err.getLineNumber () +
        ", uri " + err.getSystemId ());
    System.out.println(" " + err.getMessage ());
}
catch (SAXException e)
{
    e.printStackTrace();
}
catch (Throwable t)
{
    t.printStackTrace();
}
}
```

javac Simpleparser.java

XML

```
<?xml version="1.0"?>
<pizza>
  <topping extracheese="yes">Pepperoni</topping>
  <price>12.99</price>
  <size>large</size>
</pizza>
```

pizza.xml

```
java Simplepaser pizza.xml
```

가

```
<?xml version="1.0" encoding="UTF-8"?>
<pizza>
  <topping extracheese="yes">Pepperoni</topping>
  <price>12.99</price>
  <size>large</size>
</pizza>
```

```
..      ??
      XML      price      prices
      error Message가
```

\*\* Parsing error, line 4, uri file:/C:/jdk1.3/pizza.xml

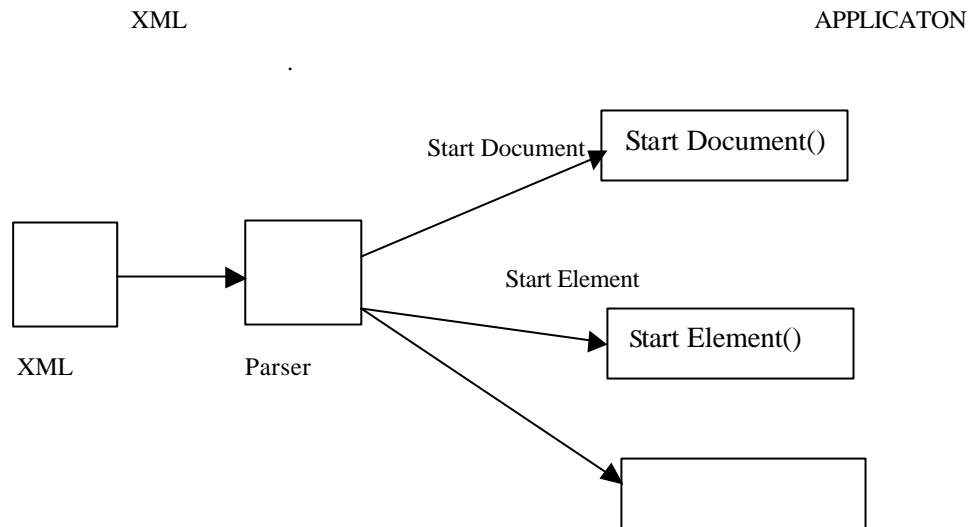
Next character must be ">" terminating element "price".

가??? ^^;

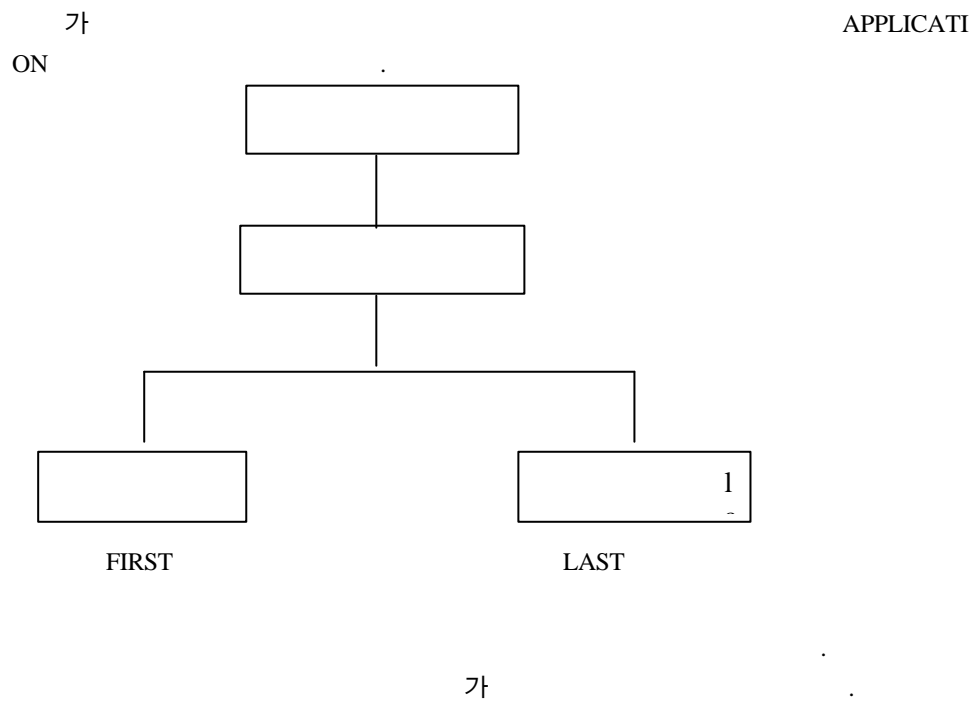
## SAX & DOM

```
XML
  (Tree)
  (Event)
  가
  W3C
  DOM
  가
  , W3C
  S
  AX
```

1) SAX (Simple API for XML) –



2) DOM (Document Object Model)–



**XML JSP**

XML JSP  
XML JSP

### 1. XML vs DB

DB XML 가 가  
DB DB XML  
SQL XML  
가 가 SQL DB XML

### 2. XML JSP

XML 가

```
<picture>  
<title>Alex On The Beach</title>  
<date>1999-08-08</date>  
<caption>Trying in vain to get a tan</caption>  
<image>  
<src>alex-beach.jpg</src>  
<width>340</width>  
<height>200</height>  
</image>  
<thumbnails>  
<image>  
<src>alex-beach-sm.jpg</src>  
<width>72</width>  
<height>72</height>  
</image>  
<image>  
<src>alex-beach-med.jpg</src>  
<width>150</width>  
<height>99</height>  
</image>  
</thumbnails>  
</picture>
```

3가 (small,medium,large)

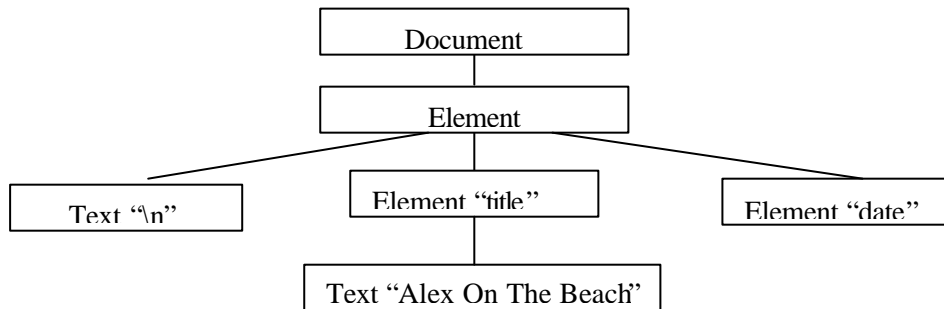
XML JSP 가 가

DOM : XML DOM

XSL:XML HTML XSL  
Xpath: XML Resin Xpath

DOM XML  
XML DOM DOM

Document Node  
--> Element Node "picture"  
--> Text Node "\n " (whitespace)  
--> Element Node "title"  
--> Text Node "Alex On The Beach"  
--> Element Node "date"  
--> ... etc.



Element Text 가 ,

--> Text Node "\n " (whitespace)

text  
DOM DOM API  
DOMUtil root(picture) element title  
text content

```
Document doc = DOMUtils.xml4jParse(picturefile);
Element nodeRoot = doc.getDocumentElement();
Node nodeTitle = DOMUtils.getChild(nodeRoot, "title");
String title = (nodeTitle == null) ? null :
```

```
DOMUtils.getTextValue(nodeTitle);
```

```
image
```

```
Node nodeImage = DOMUtils.getChild(nodeRoot, "image");  
Node nodeSrc = DOMUtils.getChild(nodeImage, "src");  
String src = DOMUtils.getTextValue(nodeSrc);
```

```
width height
```

```
JSP
```

```
<table bgcolor="#FFFFFF" border="0" cellspacing="0" cellpadding="5">  
<tr>  
<td align="center" valign="center">  
" height="<%=height%>"  
border="0" alt="<%=src%>"></td>  
</tr>  
</table>
```

## XML

### 1. Portable Data vs Portable Code

Java Portable . , Unicode Machine, OS  
XML 가 . XML Java  
XML-Dev mailing list SAX(Simple API for XML)  
JAVA . SAX  
JAVA  
SAX . , Java 가  
XML . , XML Jav  
a portable XML portable code Java code  
가 .

### 2. Object Oriented

XML Document, Element, Attribute  
XML 가  
W3C DOM  
Parser DOM  
Java Java XML .

XML  
Professional XML APPLICATION ( wrox ) : XML  
XML  
.^^;  
Java Server Programming (wrox ) : XML  
. XML

<http://www.xml.com> , <http://www.oasis-open.org/cover/>

XML 가  
가 .

[http://www.javaworld.com/javaworld/jw-03-2000/jw-0331-ssj-jspxml\\_p.html](http://www.javaworld.com/javaworld/jw-03-2000/jw-0331-ssj-jspxml_p.html)

<http://www.enhydra.org/>

XML 가 . XML