Linked Data & Semantic Web Technology

COM556
SEMANTIC WEB TECHNOLOGIES

Week 2
eXtensible Markup Language (XML)

Assist. Prof. Dr. Melike Şah Direkoğlu

Acknowledgements:
Dr. Myungjin Lee's lecture notes from Linked Data and Semantic We b Technology (Korea) was used

XML

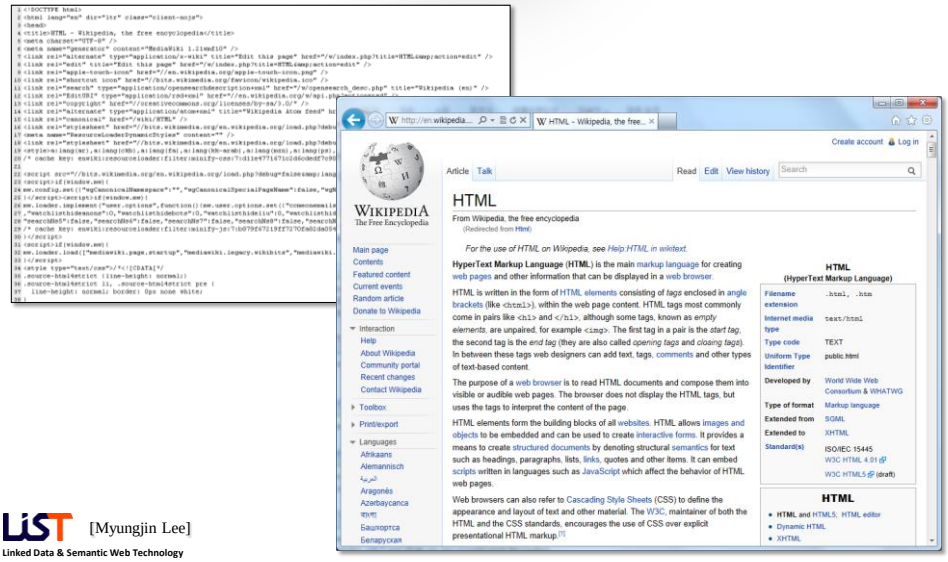
- What is XML?
- Basic of XML Document
- How to make XML Document
- XML related Recommendations
- XML Applications

XML

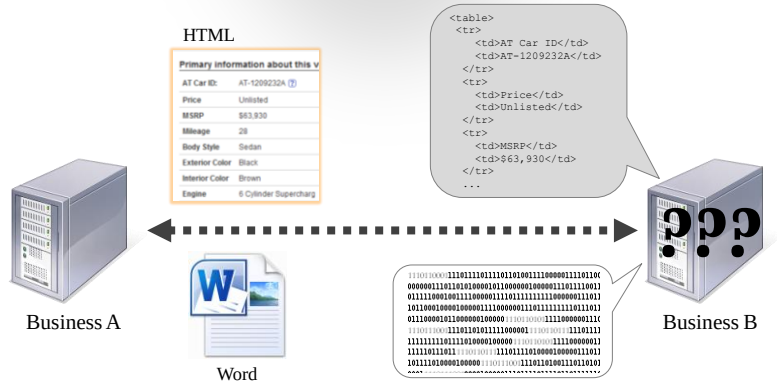
- **What is XML?**
- Basic of XML Document
- How to make XML Document
- XML related Recommendations
- XML Applications

HTML (HyperText Markup Language)

- designed to display data, with focus on how data looks



What is the problem?

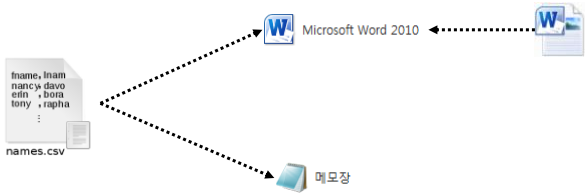


- to exchange information effectively:
  - use text file, not binary file
  - use structured document format

Binary File and Text File

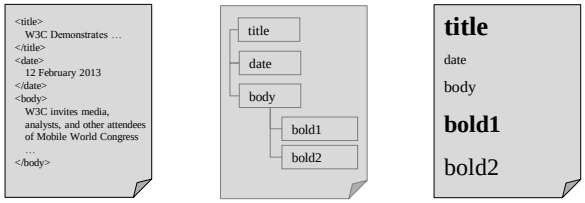
- Binary File
  - file encoded in binary form for computer storage and processing purposes
  - Problems
    - **Platform dependence**
    - Firewalls
    - Hard to debug
    - Inspecting the file is difficult

- Text File
  - file is structured as a sequence of lines of standardized electronic text
  - **Platform independence**



Structured Document

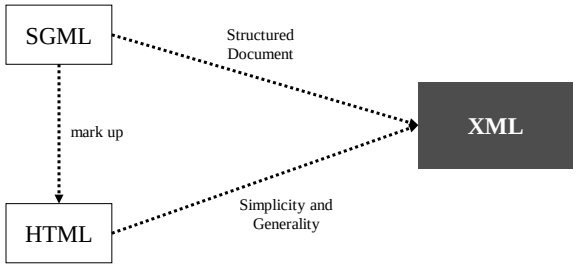
- When the whole or parts of an electronic document is **embedded (encoded) with various structural meanings according to a schema**, it is called a structured document



Content **Structure** **Presentation**

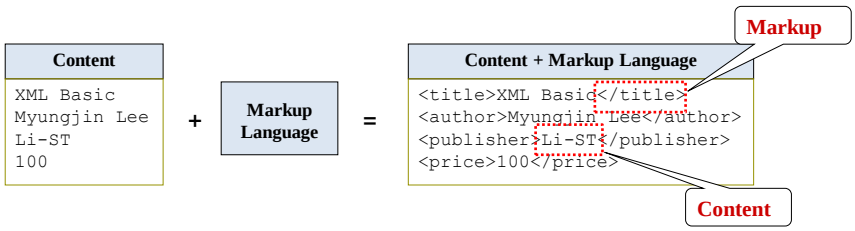
XML

- **XML** = E**x**tensible + **M**arkup **L**anguage
- **a markup language** that defines a set of rules for encoding documents in a format that is **both human-readable and machine-readable to be served, received, and processed on the Web**
- subset of SGML(Standard Generalized Markup Language)
- W3C Recommendation



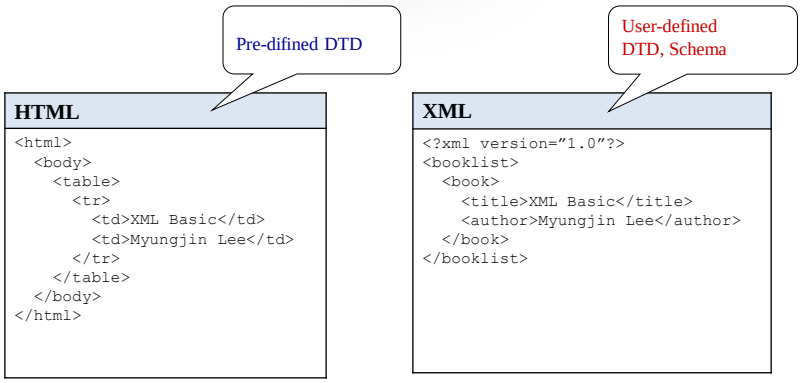
Markup Language

- Is a modern system for **annotating a document** in a way that is syntactically distinguishable from the text
- Annotation – adding useful comments/structure into the text
- History
  - evolved from the "marking up" of manuscripts, i.e., the revision instructions by editors, traditionally written with a blue pencil on authors' manuscripts



Extensible

- You can use to create **your own markup language** using Document Type Definition (DTD). It is called Schema.



Design Goal of XML

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.

Advantage of XML

- Simplicity
- Generality
- Usability
- Compatibility
- Platform independence
- Extensibility
- Rendering as different formats
- Improvement in search system

Summary

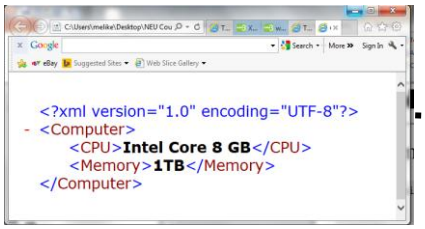
	HTML	SGML	XML
Appearance	1991	1986	1998
Purpose	Information Representation	Information Description	Information Description on the Web
Grammar	looseness	strict complex and massive	strict simple and concise
Tag	pre-defined	user-define	user-define
Information Retrieval	text-base	tree-base	tree-base
Reusability	difficult	easy	easy

XML

- What is XML?
- **Basic of XML Document**
- How to make XML Document
- XML related Recommendations
- XML Applications

Simple XML Document

- How to write XML Document
  - Open a text editor, such as Notepad
  - Save your XML file as "~.xml"



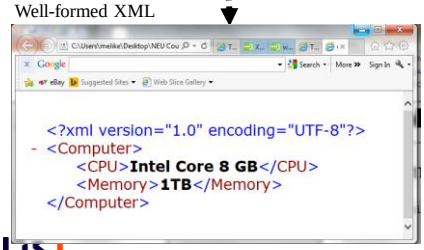
Simple.xml

Simple XML Document

- Open your XML file using a web browser, such as IE

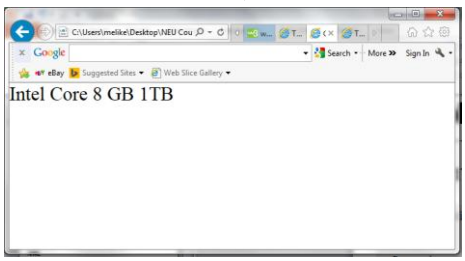
```
<?xml version="1.0" encoding="UTF-8"?>

<Computer>
      <CPU>Intel Core 8 GB</CPU>
      <Memory>1TB</Memory>
</Computer>
```

Well-formed XML



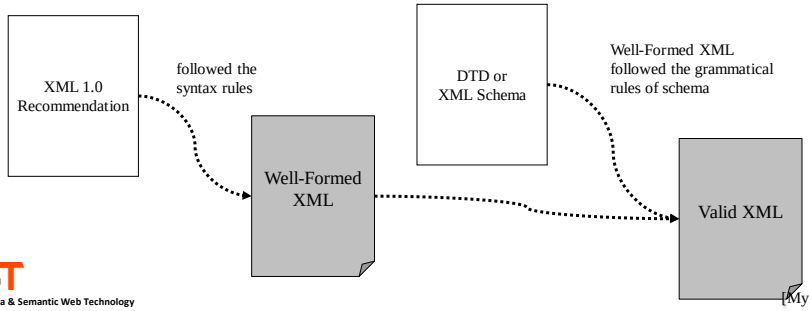

```
<?xml version="1.0" encoding="UTF-8"?>

<Computer>
      <CPU>Intel Core 8 GB</CPU>
      <Memory>1TB</Mem>
</Computer>
```

Not well-formed XML

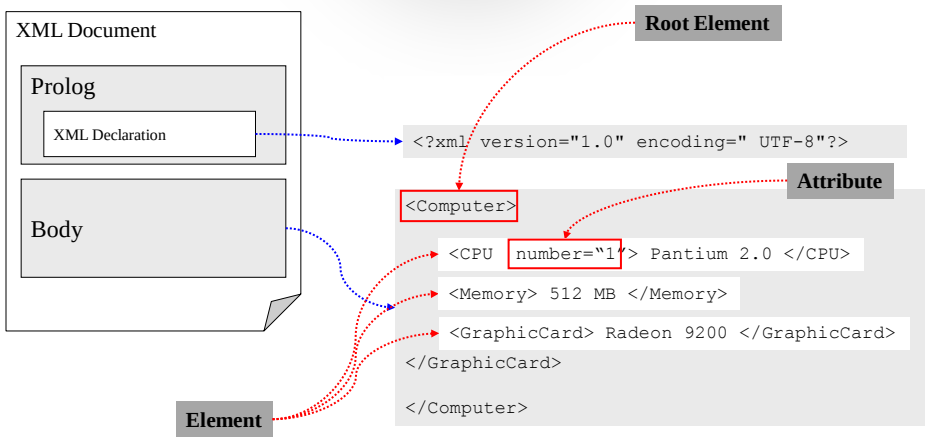


Well-formed XML and Valid XML

- Well-formed XML
  - a document that adheres to the syntax rules specified by the XML 1.0 specification in that it must satisfy both physical and logical structures
- Valid XML
  - in addition to being well-formed, XML contains a reference to a Document Type Definition (DTD), and that its elements and attributes are declared in that DTD and follow the grammatical rules for them that the DTD specifies.

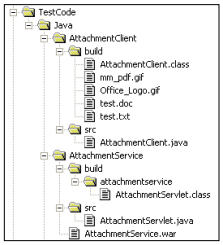


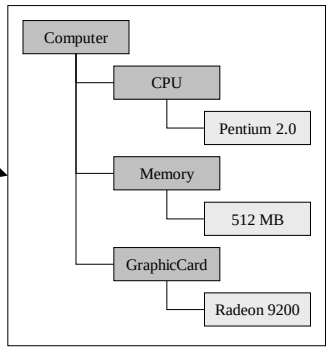
Physical Structure of XML Document



Logical Structure of XML Document

```
<?xml version="1.0" encoding=" UTF-8"?>
<Computer>
    <CPU>Pentium 2.0</CPU>
    <Memory>512 MB</Memory>
    <GraphicCard>Radeon 9200</GraphicCard>
</Computer>
```





Logical View of XML Document
A Tree

XML

- What is XML?
- Basic of XML Document
- **How to make XML Document**
- XML related Recommendations
- XML Applications

XML Declaration

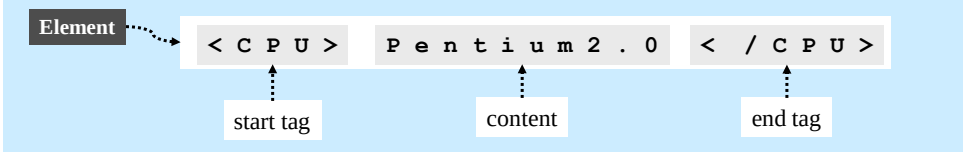
- XML documents may begin by declaring some information about themselves.

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

- Attributes of XML Declaration
  - version
    - mandatory
    - to specify the version of XML document
  - encoding
    - optional
    - to specify the encoding style of XML document
    - default value – "utf-8"
  - standalone
    - optional
    - to specify whether if the XML document is linked to external markup declaration
    - default value – "no"

Element

- Basic component of XML document



- Making Rules
  - Element consists of start tag, optional content, and end tag.
  - case-sensitive
  - Content is optional. – empty element
  - Elements must be ended with the end tag in correct order.

Element Examples

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

2. <Book>
3.    <Title>XML</Title>
4.    <Title>XML wrong
5.    <Title>XML</title> wrong
6.    <Publish></Publish>
7.    <Publish/>
8.    <Author><name>Myungjin</name></Author>
9.    <Author><name>Myungjin</Author></name>
   wrong

10.</Book>
```


Root Element

- There is exactly one element, called the root, or document element.

```
<?xml version="1.0"?>
<Book>
   <Title>XML</Title>
   <Author>Myungjin Lee</Author>
</Book>
```

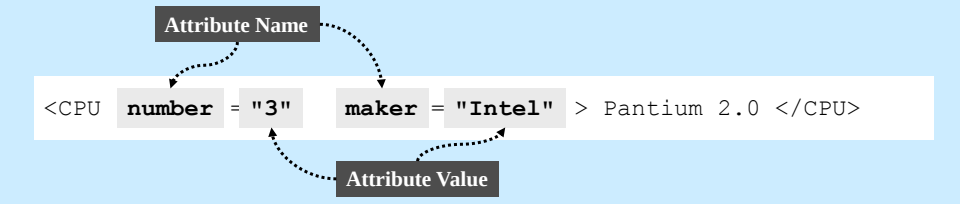
```
<?xml version="1.0"?>
<Title>XML</Title>
<Author>Myungjin Lee</Author>
```

```
<?xml version="1.0"?>
This is unvalid XML document.
```

Attribute

- XML elements can have zero and more attributes in the start tag.

- Making Rules
  - An attribute consist of name and value pair.
  - Attributes must be quoted.



Attribute Examples

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

2. <Book>
3. <Title number="3">XML</Title>
4. <Title number='3'>XML</Title>
5. **<Title number>XML</Title> wrong**
6. **<Title number=3>XML</Title> wrong**
7. </Book>

Naming Rules of Element and Attribute

- Names can contain letters, numbers, and other characters.
- Names must not start with a number or punctuation character.
- Names cannot contain spaces.
- Names must not start with the letters xml (or XML)
- Authors should not use the colon in XML names except for namespace purposes

Comments

- Making Rule
  - Comments start with "<!--" and end with "-->".

```
<!-- This is a comment, and it is not processed by XML parser -->
```

  - Comments may appear anywhere in a document outside other markup.
  - the string " --" (double-hyphen) MUST NOT occur within comments.

Special Characters and CDATA Section

- XML has a special set of characters that cannot be used in normal XML strings.
  - & - &
  - < - <
  - > - >
  - " - "
  - ' - '

- CDATA Section
  - a section of element content that is marked for the parser to interpret as only character data
  - CDATA sections begin with the string " <![CDATA[" and end with the string "]]> ".

```
<![CDATA[      Use Special Characters        ]]>
```


CDATA Section Examples

1. <?xml version="1.0" encoding="UTF-8"?>
2. <Code>
3. **<Line>if (a > b & b < c)</Line> wrong**
4. <Line>if (a > b & b < c)</Line> **OK**
5. <Line><![CDATA[if (a > b & b < c)]]></Line> **OK**
6. </Code>

Example of XML Document

ISBN	X89-89984-06
Title	XML and XML Web Services
Author	Myungjin Lee
Page	1000
Publish Data	2005. 11. 5.
Price	25000 won

```
1.  <?xml version="1.0" standalone="yes"?>

2.  <Book ISBN="X7-82219821-7">
3.      <Title>XML and XML Web Services</Title>
4.      <Author>Myungjin Lee</Author>
5.      <Page>1000</Page>
6.      <Publish year="2005" month="11" date="5" />
7.      <Price unit="won">25000</Price>
8.  </Book>
```


Creating Your XML document (Simple Note)

- Create an XML document that has a root node <note>
- Within the <note>, you will have the following XML tags
  - <to>
  - <from>
  - <heading>
  - <body>

- <?xml version="1.0" encoding="UTF-8"?>
- <note>
- <to>Tove</to>
- <from>Jani</from>
- <heading>Reminder</heading>
- <body>Don't forget to bring your umbrella!</body>
- </note>

- http://www.w3schools.com/xml/xml_examples.asp for more examples

Creating Your XML document (Food Menu)

- Create a food menu for breakfast.
- You could have <breakfast_menu> as your root node.
- <food> tag to describe various foods.

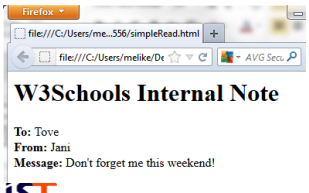
```
<?xml version="1.0" encoding="UTF-8"?>
<breakfast_menu>
    <food>
        <name>Belgian Waffles</name>
        <price>$5.95</price>
        <description>Belgian Waffles with plenty of real maple syrup</description>
        <calories>650</calories>
    </food>
    <food>
        <name>Strawberry Belgian Waffles</name>
        <price>$7.95</price>
        <description>Light Belgian waffles covered with strawberries and cream</description>
        <calories>900</calories>
    </food>
</breakfast_menu>
```


XML Parser

- All modern browsers have a built-in XML parser.
- An XML parser **converts an XML document into an XML Document Object Model (DOM)** - which can then be manipulated with a scripting language such as JavaScript.
- A DOM (Document Object Model) defines a **standard way for accessing and manipulating documents (HTML DOM)**.
- The XML DOM defines a **standard way for accessing and manipulating XML documents**.
- The XML DOM views an XML document as a **tree-structure**.
- All elements can be accessed through the DOM tree. Their content (text and attributes) can be **modified or deleted, and new elements can be created**. The elements, their text, and their attributes are all known as nodes.

Loading an XML Document and Manupulating with XLM DOM and Javascript

- Security issues when loading a local file. Use Firefox Web browser for testing or a file in a local Web server.




```
<html>
<body>
<h1>W3Schools Internal Note</h1>
<div>
<b>To:</b> <span id="to"></span><br />
<b>From:</b> <span id="from"></span><br />
<b>Message:</b> <span id="message"></span>
</div>

<script>
 if (window.XMLHttpRequest)
   {// code for IE7+, Firefox, Chrome, Opera, Safari
   xmlhttp=new XMLHttpRequest();   }
 else
   {// code for IE6, IE5
 xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");   }
 xmlhttp.open("GET","note.xml",false);
 xmlhttp.send();
 xmlDoc=xmlhttp.responseXML;
 document.getElementById("to").innerHTML=
 xmlDoc.getElementsByTagName("to")[0].childNodes[0].nodeValue;
 document.getElementById("from").innerHTML=
 xmlDoc.getElementsByTagName("from")[0].childNodes[0].nodeValue;
 document.getElementById("message").innerHTML=
 xmlDoc.getElementsByTagName("body")[0].childNodes[0].nodeValue;
</script>
</body></html>
```


XML DOM (Document Object Model)

- Read the tutorial from http://www.w3schools.com/dom/

XML DOM Properties

These are some typical DOM properties:

- x.nodeName - the name of x
- x.nodeValue - the value of x
- x.parentNode - the parent node of x
- x.childNodes - the child nodes of x
- x.attributes - the attributes nodes of x

Note: In the list above, x is a node object.

XML DOM Methods

- x.getElementsByTagName(*name*) - get all elements with a specified tag name
- x.appendChild(*node*) - insert a child node to x
- x.removeChild(*node*) - remove a child node from x

Note: In the list above, x is a node object.

Example

The JavaScript code to get the text from the first <title> element in books.xml:

txt=xmlDoc.getElementsByTagName("title")[0].childNodes[0].nodeValue

After the execution of the statement, txt will hold the value "Everyday Italian"

Explained:

- **xmlDoc** - the XML DOM object created by the parser.
- **getElementsByTagName("title")[0]** - the first <title> element
- **childNodes[0]** - the first child of the <title> element (the text node)
- **nodeValue** - the value of the node (the text itself)

XML DOM (Cont.)

Accessing Nodes

You can access a node in three ways:

1. By using the getElementsByTagName() method
2. By looping through (traversing) the nodes tree.
3. By navigating the node tree, using the node relationships.

The getElementsByTagName() Method

getElementsByTagName() returns all elements with a specified tag name.

Syntax

```
node.getElementsByTagName("tagname");
```

Example

The following example returns all <title> elements under the x element:

```
x.getElementsByTagName("title");
```

Note that the example above only returns <title> elements under the x node. To return all <title> elements in the XML document use:

```
xmlDoc.getElementsByTagName("title");
```

where xmlDoc is the document itself (document node).

XML DOM (Cont.)

DOM Node List Length

The length property defines the length of a node list (the number of nodes).

You can loop through a node list by using the length property:

Example

```
var xmlDoc=loadXMLDoc("books.xml");
var x=xmlDoc.getElementsByTagName("title");

for (i=0;i<x.length;i++)
  {
  document.write(x[i].childNodes[0].nodeValue);
  document.write("<br>");
  }
```

Node Types

The **documentElement** property of the XML document is the root node.

The **nodeName** property of a node is the name of the node.

The **nodeType** property of a node is the type of the node.

You will learn more about the node properties in the next chapter of this tutorial.

Node type	NodeType
Element	1
Attribute	2
Text	3
Comment	8
Document	9

XML DOM (Cont.)

- Traversing Nodes

Example

```
var xmlDoc=loadXMLDoc("books.xml");

//the x variable will hold a node list
var x=xmlDoc.getElementsByTagName('title');

for (i=0;i<x.length;i++)
  {
  document.write(x[i].childNodes[0].nodeValue);
  document.write("<br>");
  }
```

Output:

```
Everyday Italian
Harry Potter
XQuery Kick Start
Learning XML
```


XML

- What is XML?
- Basic of XML Document
- How to make XML Document
- **XML related Recommendations**
- XML Applications

XML Namespace

- Problem of XML
  - pose problems of recognition and collision

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<html>
  <body>
    <p>Welcome to my Health Resource</p>
  </body>

  <body>
    <height>6ft</height>
    <weight>155 lbs</weight>
  </body>
</html>
```

- XML Namespace
  - for providing uniquely named elements and attributes in an XML document
  - If each vocabulary is given a namespace, the ambiguity between identically named elements or attributes can be resolved.

How to use XML Namespace

- XML Namespace Declaration
  - using ‘xmlns’ attribute

```
<tagname  xmlns: prefix  =  namespace URI  >
```

- Applying Namespaces
  - element or attribute name starts with the prefix and ":".

```
<  prefix:  tagname> ... </  prefix:  tagname>
```

XML Namespace Example

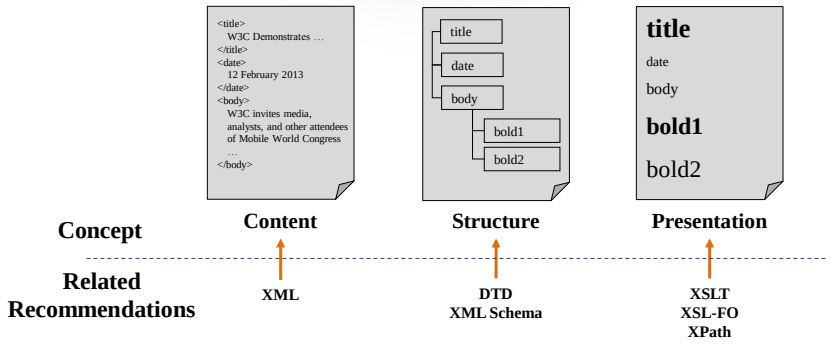
```
1. <?xml version="1.0" encoding="euc-kr"?>
2.
3. <com:Computer    xmlns:com="http://www.computer.com">
4.       <com:CPU>Pantium 2.0</com:CPU>
5.       <com:Memory>512 MB</com:Memory>
6.       <com:GraphicCard>
7.               Radeon 9200
8.       </com:GraphicCard>
9. </com:Computer>
```


Default Namespace

- **unprefixed** declaration of Namespace
- It applies to all unprefixed element names within its scope.

```
1.  <?xml version="1.0" encoding="euc-kr"?>
2.
3.  <Computer xmlns="http://www.computer.com">
4.      <CPU>Pantium 2.0</CPU>
5.      <Memory>512 MB</Memory>
6.      <GraphicCard>
7.              Radeon 9200
8.      </GraphicCard>
9.  </Computer>
```

XML related Technologies



How to Define the Structure of XML Document

- DTD (Document Type Definition)
  - a set of **markup declarations** that define a document type for an SGML-family markup language

- XML Schema
  - to express a set of rules to which an XML document must conform in order to be **considered 'valid' according to that schema**
  - generally to use a **prefix `:xsd`** for XML Namespace
  - Comparing with DTD
    - XML syntax
    - various datatypes and user-define datatype
    - various content model and occurrence constraint
    - supporting XML Namespace

DTD Example

```
1.  <?xml version="1.0" encoding="euc-kr"?>

2.  <!DOCTYPE Computer [
3.  <!ELEMENT Computer (CPU, Memory, GraphicCard)>
4.  <!ELEMENT CPU (#PCDATA)>
5.  <!ELEMENT Memory (#PCDATA)>
6.  <!ELEMENT GraphicCard (#PCDATA)>
7.  ]>

8.  <Computer>
9.      <CPU>Pantium III</CPU>
10.     <Memory>512 MB</Memory>
11.     <GraphicCard>Radeon 9200</GraphicCard>
12. </Computer>
```


XML Schema Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                    elementFormDefault="qualified">
   <xsd:element name="car">
      <xsd:complexType>
         <xsd:sequence>
            <xsd:element name="name" type="xsd:string" />
            <xsd:element name="CC" type="xsd:integer" />
         </xsd:sequence>
      </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

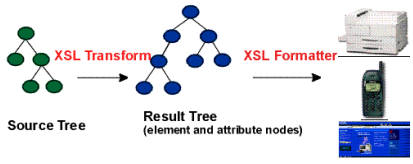
XML Schema Document

```
<?xml version="1.0"?>
<car>
   <name>TVlon</name>
   <CC>2000</CC>
</car>
```

XML Document

Stylesheet for XML

- XPath (XML Path Language)
  - a query language for selecting nodes from an XML document
- XSLT (Extensible Stylesheet Language Transformations)
  - a language for transforming XML documents into other XML documents
- XSL-FO (XSL Formatting Objects)
  - a markup language for XML document formatting which is most often used to generate PDFs

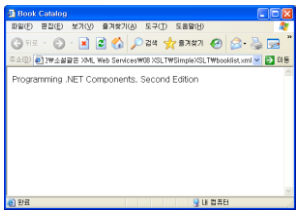


Result XML tree is the result of XSLT processing.

XSLT and XPath Example

```
1.  <?xml version="1.0" encoding="euc-kr" standalone="no"?>

2.  <?xml-stylesheet type="text/xsl" href="booklist.xsl"?>

3.  <book isbn="h0-596-00762-0">
4.    <name outpage="ProgrammingXML.gif">
5.          Programming XML
6.    </name>
7.  </book>
```

```
1.<?xml version="1.0" encoding="euc-kr"?>

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

5.  <xsl:output method="html" />

6.  <xsl:template match="/">
7.    <HTML>
8.      <BODY>
9.        <xsl:apply-templates select="book" />
10.      </BODY>
11.   </HTML>
12. </xsl:template>
13.
14. <xsl:template match="book">
15.   <xsl:value-of select="name" />
16. </xsl:template>

17.</xsl:stylesheet>
```



XML

- What is XML?
- Basic of XML Document
- How to make XML Document
- XML related Recommendations
- **XML Applications**

MathML (Mathematical Markup Language)

- an application of XML for describing mathematical notations and capturing both its structure and content

```
<math>
  <mrow>
    <mi>a</mi>
    <mo>&#x2062; </mo>
    <msup>
      <mi>x</mi>
      <mn>2</mn>
    </msup>
    <mo>+</mo>
    <mi>b</mi>
    <mo>&#x2062; </mo>
    <mi>x</mi>
    <mo>+</mo>
    <mi>c</mi>
  </mrow>
</math>
```

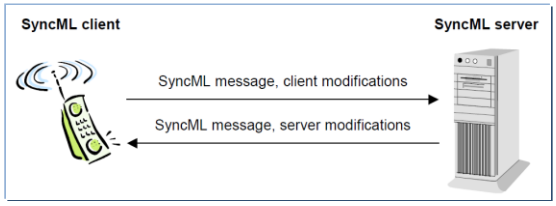
SVG (Scalable Vector Graphics)

- an **XML-based vector image format** for two-dimensional graphics that has support for interactivity and animation

```
<svg width="100%" height="100%" version="1.1"
                xmlns="http://www.w3.org/2000/svg">
        <rect width="300" height="100"
                style="fill:rgb(0,0,255);
                        stroke-width:1;stroke:rgb(0,0,0)"/>
</svg>
```


SyncML (Synchronization Markup Language)

- the former name for **a platform-independent information synchronization standard**
- to offer an open standard as a replacement for existing data synchronization solutions, which have mostly been somewhat vendor-, application- or operating system specific



XBRL (Extensible Business Reporting Language)

- a freely available and global standard for **exchanging business information** with the expression of semantic meaning commonly required in business reporting

- According to the Financial Times, "the Securities and Exchange Commission (SEC) in the US, the UK's HM Revenue & Customs (HMRC), and Companies House in Singapore have begun to require companies to use it, and other regulators are following suit.".

References

- Doug Tidwell, “Tutorial: Introduction to XML”, 1999.
- http://www.slideshare.net/pohjus/introduction-to-xml-presentation
- http://en.wikipedia.org/wiki/Structured_document
- http://en.wikipedia.org/wiki/Markup_language
- http://en.wikipedia.org/wiki/Xml
- http://en.wikipedia.org/wiki/List_of_XML_markup_languages
- http://en.wikipedia.org/wiki/Xml_parser
- http://en.wikipedia.org/wiki/Well-formed_document
- http://en.wikipedia.org/wiki/CDATA
- http://en.wikipedia.org/wiki/Document_Type_Definition
- http://en.wikipedia.org/wiki/Xml_namespace
- http://en.wikipedia.org/wiki/Xpath
- http://en.wikipedia.org/wiki/Xslt
- http://en.wikipedia.org/wiki/XSL-FO
- http://www.w3.org/TR/xsl/
- http://en.wikipedia.org/wiki/List_of_XML_markup_languages
- http://en.wikipedia.org/wiki/Mathml
- http://en.wikipedia.org/wiki/Svg
- http://en.wikipedia.org/wiki/Syncml
- http://en.wikipedia.org/wiki/XBRL