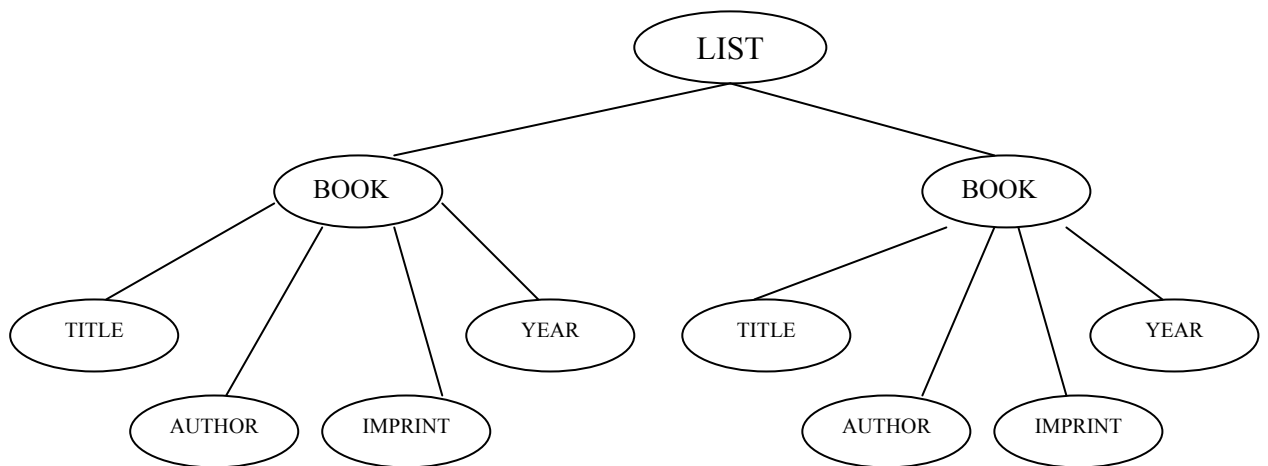


## XML

ภาษา XML (Extensible Markup Language) เป็นภาษา markup ตัวหนึ่งเหมือนกับ HTML แต่ภาษา HTML มี Element ที่แน่นอนไม่สามารถเปลี่ยน element ได้เช่น <HEAD> จะเปลี่ยนหรือกำหนด Element ใหม่ เช่น <BOOK> ไม่ได้ ภาษา HTML ประกอบด้วย Tag เปิด และ tag ปิด เช่น <HEAD>.....</HEAD> ข้อจำกัดต่างๆเช่น การจัดเรียงฐานข้อมูล การสร้างเอกสารแบบโครงสร้างต้นไม้ (มี element ซ้อน element ) ไม่สามารถทำได้ จึงเกิดภาษา XML เพื่อแก้ปัญหาและข้อจำกัดดังกล่าว

XML ย่อมาจาก Extensible Markup Language หมายถึงการขยายการใช้งานของ Markup

<pre>&lt;?xml version="1.0"?&gt; &lt;LIST&gt; &lt;BOOK&gt;   &lt;TITLE&gt;XML step by stem&lt;/TITLE&gt;   &lt;AUTHOR&gt; Young, Michael J&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;Redmond, WA : Microsoft Pr.&lt;/IMPRINT&gt;   &lt;YEAR&gt;2000&lt;/YEAR&gt; &lt;/BOOK&gt; &lt;BOOK&gt;   &lt;TITLE&gt;XML: the complete reference&lt;/TITLE&gt;   &lt;AUTHOR&gt; Williamson, Heather&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;New York : Osborne/McGraw-Hill&lt;/IMPRINT&gt;   &lt;YEAR&gt;2001&lt;/YEAR&gt; &lt;/BOOK&gt; &lt;/LIST&gt;</pre>	<p>XML declaration องค์ประกอบเอกสาร &lt;BOOK&gt;&lt;TITLE&gt;&lt;AUTHOR&gt;&lt;IMPRINT&gt; &lt;YEAR&gt;เป็น องค์ประกอบข้อมูล XML step by step เป็นค่าของข้อมูล</p>
--	--



<pre>&lt;?xml version="1.0"?&gt; &lt;!-- First line call XML Declaration --&gt;  &lt;!-- Comment --&gt; &lt;!-- Prolog can have .... Document Type Declaration(DTD) --&gt; &lt;!-- Prolog can have .... Processing Instruction(PI) --&gt; &lt;!-- End Prolog Section --&gt;  &lt;!-- Begin Document Element(Root Element) Section --&gt; &lt;LIST&gt;  &lt;BOOK&gt;   &lt;TITLE&gt;XML step by stem&lt;/TITLE&gt;   &lt;AUTHOR&gt; Young, Michael J&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;Redmond, WA : Microsoft Pr.&lt;/IMPRINT&gt;   &lt;YEAR&gt;2000&lt;/YEAR&gt; &lt;/BOOK&gt;  &lt;BOOK&gt;   &lt;TITLE&gt;XML: the complete reference&lt;/TITLE&gt;   &lt;AUTHOR&gt; Williamson, Heather&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;New York : Osborne/McGraw-Hill&lt;/IMPRINT&gt;   &lt;YEAR&gt;2001&lt;/YEAR&gt; &lt;/BOOK&gt; &lt;/LIST&gt;</pre>	<p>เอกสาร xml ประกอบด้วย 2 ส่วน คือ</p> <ol style="list-style-type: none"> <li>1. Prolog ประกอบด้วยxml declaration ประกาศว่าเป็นเอกสาร XML &lt;1... comment → ข้อความที่อยู่ในเอกสารจะไม่ประมวลผล นอกจากนี้ยังมี document type หรือ DTD และ Processing Instruction หรือ PI</li> <li>2. Documentant ในส่วนของ document จะต้องเริ่มต้นด้วย root element เช่น &lt;LIST&gt; และต้องมี tag ปิดคือ &lt;/LIST&gt; ใน &lt;LIST&gt; สามารถจะมี element ซ้อนอยู่ได้เช่น &lt;BOOK&gt; และใน M&lt;BOOK&gt;จะมี element อื่นซ้อนอยู่ได้ เช่น &lt;TITLE&gt;&lt;AUTHOR&gt;&lt;IMPRINT&gt;&lt;YEAR&gt;</li> <li>3. processing instruction</li> </ol>
--	--

### กฎพื้นฐาน

<pre>&lt;?xml version="1.0"?&gt; &lt;BOOK&gt;   &lt;TITLE&gt;XML step by step &lt;/TITLE&gt;   &lt;AUTHOR&gt; Young, Michael J&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;Redmond, WA : Microsoft Pr.&lt;/IMPRINT&gt;   &lt;YEAR&gt;2000&lt;/YEAR&gt; &lt;/BOOK&gt; &lt;BOOK&gt;   &lt;TITLE&gt;XML: the complete reference&lt;/TITLE&gt;   &lt;AUTHOR&gt; Williamson, Heather&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;New York : Osborne/McGraw-Hill&lt;/IMPRINT&gt;   &lt;YEAR&gt;2001&lt;/YEAR&gt; &lt;/BOOK&gt;</pre>	<p>1. เอกสาร xml ต้องมี element บนสุดเพียง element เดียว ซึ่งเรียกว่า root element</p>
<pre>&lt;BOOK&gt;&lt;TITLE&gt;XML: Step by Step&lt;/BOOK&gt;&lt;/TITLE&gt; ↓ &lt;BOOK&gt;&lt;TITLE&gt;XML: Step by Step&lt;/TITLE&gt;&lt;/BOOK&gt;</pre>	<p>2. แต่ละ element ต้องซ้อนกันอย่างถูกต้อง ภายใน element</p>
<pre>&lt;BOOK&gt;&lt;TITLE&gt;XML: Step by Step&lt;/BOOK&gt;</pre>	<p>แต่ละ element ต้องเปิดและปิด tag ให้ถูกต้องยกเว้น element พิเศษเช่น empty element ไม่ต้องมี tag ปิด</p>
<pre>&lt;TITLE&gt;XML: Step by Step&lt;/Title&gt;</pre>	<p>ชื่อของ element เป็น case-ensitive</p>
<pre>&lt;TITLE&gt;XML: Step by Step&lt;/BOOK&gt; ↓ &lt;TITLE&gt;XML: Step by Step&lt;/TITLE&gt;</pre>	<p>ชื่อของ element ใน tag เริ่มต้นต้องตรงกับ tag ปิดท้าย</p>

## การสร้างเอกสาร XML มี 2 ชนิดคือ

1. เอกสารที่มีรูปแบบถูกต้องหรือเรียกว่า Well-formed XML Document ต้องมีกฎพื้นฐานที่บังคับ เอกสาร เรียกว่า well form document
2. เอกสารที่ถูกต้องสมบูรณ์ valid XML Document ต้องมี DTD (Document Type Declaration) ซึ่งเป็นตัวควบคุมเอกสารนั้นๆ หรือเป็นไวยากรณ์ของเอกสาร ดังนั้นองค์ประกอบต่างๆจึงแลกเปลี่ยนข้อมูลกัน ได้ โดยใช้ DTD เป็นสื่อกลาง

## การแสดงผลเอกสาร XML

1. ไม่เชื่อมโยงกับ Style Sheet การแสดงผลใช้ web browser ลักษณะการแสดงผลจะเป็นแบบต้นไม้ สีของข้อมูลจะแตกต่างกันกับ tag และ comment ก่อนที่จะแสดงผล xml จะประมวลผลก่อนเสมอ ถ้ามี tag ผิด จะไม่สามารถแสดงผลได้ และจะแสดง error
2. เชื่อมโยงกับ Style Sheet จะต้องสร้าง 2 file คือ 1. XML document และ 2. Cascading Style Sheet และ เชื่อม XML document และ CSS เข้าด้วยกันด้วยคำสั่ง

<?xml-stylesheet type="text/css" href="Book.css"?>

<pre>สร้าง file Book.css /* Book.css */ BOOK {display:block; margin-top:12pt; font-size:10pt} TITLE {display:block; font-size:12pt; font-weight:bold; font-style:italic} AUTHOR {display:block; margin-left:15pt; font-weight:bold} IMPRINT {display:none} YEAR {display:block;</pre>	
---	--

## การสร้างเอกสาร XML มี 2 ชนิดคือ

1. เอกสารที่มีรูปแบบถูกต้องหรือเรียกว่า Well-formed XML Document โดยใช้กฎพื้นฐาน

## 2. เอกสารที่ถูกต้องสมบูรณ์ โดยใช้กฎพื้นฐาน และ DTD

### XML Element

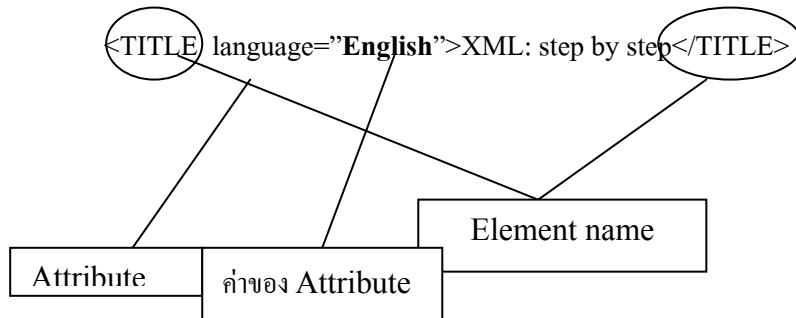
โครงสร้างของ element

1. Normal Element เป็น element ที่มีข้อมูล เช่น <TITLE>XML: Step by Step</TITLE>
2. Empty Element เป็น element ที่ไม่มีข้อมูลเช่น tag <BR> หรือ <BR/>

ข้อมูลที่อยู่ใน element มีประเภทต่างๆดังนี้

<pre>&lt;BOOK&gt;   &lt;TITLE&gt;XML step by stem&lt;/TITLE&gt;   &lt;AUTHOR&gt; Young, Michael J&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;Redmond, WA : Microsoft Pr.&lt;/IMPRINT&gt;   &lt;YEAR&gt;2000&lt;/YEAR&gt; &lt;/BOOK&gt;  &lt;BOOK&gt;   &lt;TITLE&gt;XML: the complete reference&lt;/TITLE&gt;   &lt;AUTHOR&gt; Williamson, Heather&lt;/AUTHOR&gt;   &lt;IMPRINT&gt;New York : Osborne/McGraw-   Hill&lt;/IMPRINT&gt;   &lt;YEAR&gt;2001&lt;/YEAR&gt; &lt;/BOOK&gt;</pre>	<p>1. Nested Element เป็นข้อมูลที่ซ้อนกันได้</p>
<pre>&lt;TITLE&gt;XML step by step&lt;/TITLE&gt; หรือ &lt;TITLE&gt;XML   &lt;SUBTITLE&gt;Step by Step&lt;/SUBTITLE&gt; &lt;/TITLE&gt;</pre>	<p>2. Character Data คือข้อความธรรมดาและอาจอยู่ใน Nested element ได้</p>
	<p>3. Reference</p>
<pre>&lt;![CDATA[ ปิดด้วย ]]&gt;</pre>	<p>4. CDATA สำหรับเขียนเครื่องหมายบางเครื่องหมายได้ เช่น &lt;, &amp; ใน ข้อความของ element</p>
<pre>&lt;?XML -stylesheet type="text/css" href="Book.css"?&gt;</pre>	<p>5. Processing Instruction ให้เอกสาร xml บอก application ว่าให้เชื่อมโยง style sheet กับ เพิ่มข้อมูล Book.css</p>
<pre>&lt;!-- ข้อความ --&gt;</pre>	<p>6. Comment</p>

ใน Tag ของ element สามารถมี attribute ได้เช่น



การตั้งชื่อของ element หรือ Attribute ต้องขึ้นต้นด้วยตัวอักษรหรือ \_ ไม่สามารถขึ้นต้นด้วยตัวเลข หรือ เว้นวรรค เช่น

TITLE

\_1stTITLE

A-Language

A.B.C.

### เอกสารที่ถูกต้องสมบูรณ์ valid XML Document

รูปแบบคือ <!DOCTYPE RootEleName DTD>

DTD จะประกาศในส่วนของ prolog

<pre>&lt;!DOCTYPE Example [&lt;!ELEMENT Example ANY&gt;]&gt;</pre> <p><b>สร้าง file ValidDoc.xml</b></p> <pre>&lt;?xml version="1.0"?&gt;</pre> <pre>&lt;!DOCTYPE Example [ &lt;!ELEMENT Example ANY&gt;]&gt;</pre> <pre>&lt;Example&gt;Hello, This is an Example.&lt;/Example&gt;</pre>	<p>ชื่อของ root คือ example และการประกาศใน element example มี keyword ANY สามารถมีข้อความได้ทุกชนิด</p> <p>DTD เป็นตัวกำหนดเอกสารว่าจะมี element และ Attribute ใดได้บ้าง</p>
Element Type Declaration	ประกาศชนิดของ element ว่ามีชนิดใดได้บ้าง
Attribute-list Declaration	ประกาศ attribute
Entity Declaration	ประกาศ entity
Notation Declaration	ประกาศ Notation
Processing Instruction	ภายใน DTD สามารถใส่ Processing Instruction ได้
Comment	ภายใน DTD สามารถใส่ Comment ได้

## การประกาศ element ใน DTD

<pre> &lt;!Element name Contenspec&gt;  &lt;!Element Game(#PCDATA)&gt; ตัวอย่าง &lt;?xml version="1.0"?&gt; &lt;!DOCTYPE Software [     &lt;!ELEMENT Software (Game)+&gt;     &lt;!ELEMENT Game (#PCDATA)&gt; ]&gt; &lt;Software&gt;     &lt;Game&gt;The Sim&lt;/Game&gt;     &lt;Game&gt;Black and White&lt;/Game&gt;     &lt;Game&gt;Red Alert&lt;/Game&gt; &lt;/Software&gt;         </pre>	<p>1. จะต้องมี keyword &lt;!Element.....&gt; Name คือชื่อของ element Contentspec หมายถึงภายในที่ประกาศสามารถมีอะไรบรรจุได้บ้าง</p> <p>ภายใน element software สามารถมี game อยู่ได้ และ game นั้นสามารถมี PCDATA อยู่ได้</p>
<pre> EMPTY Content &lt;!Element BR Empty&gt; &lt;BR&gt;&lt;/BR&gt; -&gt; &lt;BR/&gt;         </pre>	<p>2. เป็น element ที่ไม่ต้องมีอะไรบรรจุ</p>
<pre> ANY Content &lt;!Element BOOK ANY&gt;         </pre>	<p>3. element ใดที่ประกาศเป็น ANY นั้น ภายใน element นั้นจะบรรจุอะไรก็ได้</p>
<pre> Element Conten 1. Sequence &lt;!ELEMENT BOOK (TITLE, AUTHOR,IMPRINT,YEAR)&gt; &lt;BOOK&gt;     &lt;TITLE&gt;XML step by stem&lt;/TITLE&gt;     &lt;AUTHOR&gt; Young, Michael J&lt;/AUTHOR&gt;     &lt;IMPRINT&gt;Redmond, WA : Microsoft Pr.&lt;/IMPRINT&gt;     &lt;YEAR&gt;2000&lt;/YEAR&gt; &lt;/BOOK&gt; 2. choice &lt;!ELEMENT BOOKTYPE (Magazine  Textbook  Reference)&gt; &lt;BOOKTYPE&gt;     &lt;Magazine&gt;PC Magazine&lt;/Magazine&gt; &lt;/BOOKTYPE&gt;         </pre>	<p>4. Element Content มี 2 รูปแบบ 1. Sequence หรือ Choice Element ต้องเรียกต่อกัน</p> <p>เลือกอย่างใดอย่างหนึ่ง</p>

<p>หรือ</p> <pre>&lt;BOOKTYPE&gt;   &lt;Textbook&gt;XML step by step&lt;/Textbook&gt; &lt;/BOOKTYPE&gt; หรือ &lt;BOOKTYPE&gt;   &lt;Reference&gt;XML Reference Guide book&lt;/Reference&gt; &lt;/BOOKTYPE&gt;</pre>	
<p>Element Content</p> <pre>&lt;?xml version="1.0"?&gt; &lt;!DOCTYPE BOOK [   &lt;!ELEMENT BOOK (Author+, Title?)&gt;   &lt;!ELEMENT Author (#PCDATA)&gt;   &lt;!ELEMENT Title (#PCDATA)&gt; ]&gt; &lt;BOOK&gt;   &lt;Author&gt;Name1&lt;/Author&gt;   &lt;Author&gt;Name2&lt;/Author&gt;   &lt;Title&gt; XML step by step &lt;/Title&gt; &lt;/BOOK&gt;</pre>	<p>5. element content สามารถมีเครื่องหมายเพิ่มเติม เช่น ? + *</p> <p>? มี child element ได้ 1 ตัวเท่านั้นหรือไม่มีเลย</p> <p>+ มี child element ตั้งแต่ 1 ตัวขึ้นไป</p> <p>* มี child element กี่ตัวก็ได้หรือไม่มีเลย</p>
<p>Mixed Content</p> <pre>&lt;!ELEMENT TITLE (#PCDATA SUBTITLE)*&gt; &lt;TITLE&gt;XML   &lt;SUBTITLE&gt;Step by Step&lt;/SUBJTITLE&gt; &lt;/TITLE&gt; หรือ &lt;TITLE&gt;XML&lt;/TITLE&gt; หรือ XML</pre>	<p>6. ภายใน title มีรูปแบบ choice ให้เลือกเป็น character data หรือ subtitle</p> <p>สำหรับ * สามารถมีได้หลาย title หรือไม่มีเลย</p>

### การประกาศ attribute ใน DTD

<pre>&lt;!ATTLIST TITLE language CDATA 'ENGLISH"&gt; &lt;?xml version="1.0"?&gt; &lt;!DOCTYPE LIST [   &lt;!ELEMENT LIST (BOOK)*&gt;   &lt;!ELEMENT BOOK (TITLE, AUTHOR)&gt;   &lt;!ELEMENT TITLE (#PCDATA)&gt;   &lt;!ATTLIST TITLE language CDATA #REQUIRED&gt;   &lt;!ELEMENT AUTHOR (#PCDATA)&gt;</pre>	<p>TITLE คือ Element Name</p> <p>Language คือ Attribute Name</p> <p>CDATA คือ Attribute Type</p> <p>English คือค่า default</p> <div style="border: 1px solid black; padding: 5px; display: inline-block; margin-top: 10px;">Attribute Definition</div> <p>จากตัวอย่างมีการประกาศ element List และข้างในมี BOOK และใน BOOK จะมี TITLE และ AUTHOR</p>
---	---

<pre> ] &lt;LIST&gt;   &lt;BOOK&gt;     &lt;TITLE language = "English"&gt; XML step by step&lt;/TITLE&gt;     &lt;AUTHOR&gt; Young, Michael J &lt;/AUTHOR&gt;   &lt;/BOOK&gt; &lt;/LIST&gt; </pre>	<p>ใน TITLE จะมี Attribute</p>
--	--------------------------------

**ชนิดของ Attribute**

<p><b>String Type ---</b></p>	<p>หมายถึง CDATA</p>
<p><b>Tokenized Type</b></p> <p><b>ID</b></p> <pre> &lt;!ATTLIST ITEM Code ID #Required&gt; &lt;ITEM Code ="A27"&gt;TV&lt;/ITEM&gt; &lt;ITEM Code = "A28"&gt;FAN&lt;/ITEM&gt; </pre> <p><b>IDREF</b></p> <pre> &lt;!ATTLIST ITEM Code ID #REQUIRED GoesWith IDREF #IMPLIED&gt; &lt;ITEM Code = "A27"&gt;TV&lt;/ITEM&gt; &lt;ITEM Code = "A28"&gt; GoesWith = "A27"&gt;Fan&lt;/ITEM&gt; </pre> <p><b>IDREFS</b></p> <p><b>ENTITY</b></p> <pre> &lt;!ELEMENT IMAGE EMPTY&gt; &lt;!ATTLIST IMAGE Source ENTITY #REQUIRED&gt; &lt;IMAGE Source = "Logo"/&gt; </pre> <p><b>ENTITIES</b></p> <p><b>NMTOKEN</b></p> <pre> &lt;!ELEMENT BOOK (#PCDATA) &lt;!ATTLIST BOOK ISBN NMTOKEN #REQUIRED&gt; &lt;BOOK ISBN = "1-23456-789-0 </pre> <p><b>NMTOKENS</b></p>	<p>ID ค่าattribute ในแต่ละ item ห้ามซ้ำกัน</p> <p>IDREF ใช้ร่วมกับ ID</p> <p>IDREFS มีค่าเหมือนกับ IDREF ยกเว้น GoesWith มีได้หลายอัน</p> <p>Entity มีการประกาศ entity ก่อนถึงจะนำไปใช้ได้ Entities มีได้หลาย entity</p> <p>NMTOKEN หมายถึงในการตั้งชื่อสามารถมีตัวเลข ขึ้นต้นได้</p> <p>NMTOKENS มีได้หลายชื่อได้</p>
<p><b>Enumerated Type</b></p> <pre> &lt;!ELEMENT BOOK (#PCDATA)&gt; &lt;!ATTLIST BOOK Type (Magazine Textbook) "Textbook"&gt; &lt;BOOK Type = "Textbook"&gt;XML step by step&lt;/BOOK&gt; </pre>	<p>เป็นการกำหนดชื่อที่ชื่อใดสามารถให้เลือกได้บ้าง</p>
<pre> &lt;TITLE Language = "English"&gt;XML step by step&lt;/TITLE&gt; &lt;!ELEMENT BOOK (#PCDATA)&gt; &lt;!ATTLIST BOOK ISBN NMTOKEN #REQUIRED &lt;BOOK ISBN = "1-23456-789-0 &lt;!-- &lt;!ATTLIST BOOK Type CDATA #FIXED "Textbook"&gt; &lt;BOOK&gt;XML step by step&lt;/BOOK&gt; หรือ &lt;BOOK Type = "Textbook"&gt; XML step by step &lt;/BOOK&gt; </pre>	<p>ค่า default มี 4 ชนิดคือค่า value หรือค่าทั่วไป</p> <p>#Required หมายถึง attribute นั้นต้องมีค่าเสมอ</p> <p>#Implied จะมีค่าหรือไม่มีค่าก็ได้เช่น A27 จะใส่หรือไม่ใส่ก็ได้</p> <p>#FIXED Value เป็นค่าที่กำหนดไว้แน่นอน</p>

## External DTS Subset

การนิยาม DTD จะอยู่ภายนอกเอกสาร XML โดยตั้งชื่อเพิ่มข้อมูล.dtd รูปแบบ

```
<!DOCTYPE LIST SYSTEM "list.dtd"
```

หรือ

```
<!DOCTYPE LIST SYSTEM http://xml.com/list.dtd>
```

สร้าง file booklist.xml และ list.dtd ดังนี้

<pre>&lt;?xml version="1.0"?&gt; &lt;!DOCTYPE LIST SYSTEM "list.dtd" &gt; &lt;LIST&gt;   &lt;BOOK&gt;     &lt;TITLE language = "English"&gt; XML step by step   &lt;/TITLE&gt;   &lt;AUTHOR&gt; Young, Michael J &lt;/AUTHOR&gt;   &lt;/BOOK&gt; &lt;/LIST&gt;</pre>	<pre>&lt;!ELEMENT LIST (BOOK)*&gt; &lt;!ELEMENT BOOK (TITLE, AUTHOR)&gt; &lt;!ELEMENT TITLE (#PCDATA)&gt; &lt;!ATTLIST TITLE language CDATA #REQUIRED&gt; &lt;!ELEMENT AUTHOR (#PCDATA)&gt;</pre>
--	---

## CSS

### Font and color

<pre>{display:block; font-family: "Arial"; font-size: 12pt; font-style:normal; font-weight:normal; font-variant:saml1-caps; color=blue}</pre>	<p>Times New Roman, Western Medium, large, Italic, oblique Bold, 500 Normal</p>
---	---

### Background

<pre>/* File Name: background.css */ Set {font-size:x-large;} Set, Set1, Set2, Set3, Set4, Set5, Set6 {display:block; margin-top: 1em} Set1 {background-color:yellow} Set2 {background-color:red} Set3 {background-image:url(Leaf.bmp)} Set4 {background-image:url(smile.bmp)} Set5 {background-image:url(smile.bmp); background-repeat:repeat-x} Set6 {background-image:url(smile.bmp); background-repeat:no-repeat}</pre>	<p>ข้อมูลชุดที่1 พื้นจะเป็นสีเหลือง</p> <p>ข้อมูลชุดที่2 พื้นจะเป็นสีแดง</p> <p>ข้อมูลชุดที่3 พื้นจะเป็นลายใบไม้</p> <p>ข้อมูลชุดที่4 พื้นจะเป็นหน้ายิ้ม</p> <p>ข้อมูลชุดที่5 พื้นจะเป็นหน้ายิ้ม และการวางรูปจะวางซ้ำในแนวแกน x</p> <p>ข้อมูลชุดที่6 พื้นจะเป็นหน้ายิ้ม เพียงรูปเดียว</p>
---	---

Text spacing เปิด file text.xml

<pre> /* File Name: text.css */  Root, Left, Right, Center, Underline     {display:block}  /** letter-spacing *****/ Normal     {letter-spacing:normal} Small     {letter-spacing:.5em}  /** vertical-align *****/ Sub     {font-size: 75%;     vertical-align:sub} Super     {font-size: 75%;     vertical-align:super}  ***** text-align *****/ Left     {text-align:left} Right     {text-align:right} Center     {text-align:center}  **** text-transform *****/ Uppercase     { text-transform: uppercase} Lowercase     { text-transform: lowercase} Capitalize     { text-transform: capitalize} </pre>	<ul style="list-style-type: none"> <li>- Letter-spacing</li> <li>- Vertical align</li> <li>- Text align</li> <li>- Text transform</li> </ul> <p>การแสดงผลของช่องตัวอักษรเป็นแบบธรรมดา(Normal)</p> <p>การแสดงผลของช่องตัวอักษรเป็นแบบใหญ่(Big ) โดยกำหนดค่าเป็นตัวเลข .5em</p> <p>การแสดงผลของตัวอักษรแบบตัวห้อย(sub)</p> <p>การแสดงผลของตัวอักษรแบบตัวยกกำลัง(super)</p> <p>การแสดงผลของตัวอักษรชิดซ้าย</p> <p>การแสดงผลของตัวอักษรชิดขวา</p> <p>การแสดงผลของตัวอักษรกึ่งกลาง</p> <p>การแสดงผลของตัวอักษรแบบตัวใหญ่</p> <p>การแสดงผลของตัวอักษรแบบตัวเล็ก</p> <p>การแสดงผลของตัวอักษรเฉพาะตัวแรกเป็นตัวใหญ่</p>
--	---

Marging และ border

<pre> /* File Name: margin.css */  Root, Top, Right, Bottom, Left, Top-Right, Bottom-Left, Solid, Dotted, Dashed, Groove, Ridge, Inset, Outset, Groove-Top-Left     {display:block}  ***** Margin *****/ Top     {margin-top:3em} Right     {margin-right:3em} Bottom     {margin-bottom:3em} Left </pre>	<p>การกำหนด margin ของ top</p> <ol style="list-style-type: none"> <li>1. มีบรรทัดว่างด้านบน 3 เท่า</li> <li>2. มีช่องว่างด้านขวา</li> <li>3. มีบรรทัดว่างด้านล่าง 3 เท่า</li> </ol>
---	---

<pre> {margin-left:3em} Top-Right {margin-top:3em; margin-right:3em} Bottom-Left {margin-bottom:3em; margin-left:3em} /***** Border *****/ Solid {border-style:solid} Groove {border-style:groove} Ridge {border-style:ridge} Inset {border-style:inset} Outset {border-style:outset}  Groove-Top-Left {border-style:groove; margin-top:3em; margin-left:3em} </pre>	<p>4. มีช่องว่างด้านซ้าย</p> <p>5. ช่องว่างด้านบนและด้านขวา</p> <p>6. ช่องว่างด้านล่างและด้านซ้าย</p> <p>กรอบของข้อความ</p> <ol style="list-style-type: none"> <li>1. solid</li> <li>2. Groove</li> <li>3. Ridge</li> <li>4. Inset</li> </ol> <p>การนำกรอบของข้อความนำมารวมกับ margin</p>
--	---

### การสร้าง XSL สำหรับการแสดงผล xml

XSL มีประสิทธิภาพและยืดหยุ่นกว่า CSS

XSL สามารถเลือกข้อมูล ปรับปรุงข้อมูล จัดการข้อมูลและเข้าถึงองค์ประกอบของ XML ได้ทั้งหมด

XSL สามารถเข้าถึง attribute ได้แต่ CSS ไม่สามารถทำได้

### พื้นฐานการสร้าง

สร้างเอกสาร xml และ XSL แล้ว link กันด้วยคำสั่ง

<?XML-stylesheet type = "text/xsl" href=XSLFilePath ?> ในตอนต้นของเอกสาร xml

XSLFilePath คือที่อยู่ของเอกสาร xsl ที่สร้างขึ้น

โครงสร้าง XSL

1. XSL ต้องมี root element และ root element ต้องมี namespace
2. ภายใน root element มี template อาจมีได้มากกว่า 1 template
3. ภายใน template มี attribute ที่เรียกว่า match และค่าของ match คือ / คือ root เป็น root ทั้งหมดของเอกสาร

ตัวอย่าง      <xsl:template match="/">  
                 </xsl:template>

<pre>&lt;?xml version="1.0"?&gt;  &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"&gt;   &lt;xsl:template match="/"&gt;     &lt;H1&gt;&lt;Font color = "blue"&gt;The Book&lt;/Font&gt;&lt;/H1&gt;     &lt;I&gt;Author: &lt;/I&gt;     &lt;xsl:value-of select="BOOK/AUTHOR"/&gt;&lt;BR/&gt;     &lt;I&gt;Title: &lt;/I&gt;     &lt;xsl:value-of select="BOOK/TITLE"/&gt;&lt;BR/&gt;     &lt;I&gt;Imprint: &lt;/I&gt;     &lt;xsl:value-of select="BOOK/IMPRINT"/&gt;&lt;BR/&gt;     &lt;I&gt;Year: &lt;/I&gt;     &lt;xsl:value-of select="BOOK/YEAR"/&gt;&lt;BR/&gt;   &lt;/xsl:template&gt; &lt;/xsl:stylesheet&gt;</pre>	<p><b>The Book</b></p> <p><i>Author:</i> Young, Michael J  <i>Title:</i> XML step by stem  <i>Imprint:</i> Redmond, WA : Microsoft Pr.  <i>Year:</i> 2000</p>
---	---

การใช้งาน loop ใน xsl

```
<xsl:for-each select = "Path">
```

...

```
</xsl:for-each>
```

<pre>&lt;?xml version="1.0"?&gt;  &lt;xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl"&gt;   &lt;xsl:template match="/"&gt;     &lt;H1&gt;&lt;Font color = "blue"&gt;The Book&lt;/Font&gt;&lt;/H1&gt;     &lt;xsl:for-each select="LIST/BOOK"&gt;       &lt;I&gt;Author: &lt;/I&gt;       &lt;xsl:value-of select="AUTHOR"/&gt;&lt;BR/&gt;       &lt;I&gt;Title: &lt;/I&gt;       &lt;xsl:value-of select="TITLE"/&gt;&lt;BR/&gt;       &lt;I&gt;Imprint: &lt;/I&gt;       &lt;xsl:value-of select="IMPRINT"/&gt;&lt;BR/&gt;       &lt;I&gt;Year: &lt;/I&gt;       &lt;xsl:value-of select="YEAR"/&gt;&lt;BR/&gt;     &lt;BR/&gt;   &lt;/xsl:for-each&gt; &lt;/xsl:template&gt; &lt;/xsl:stylesheet&gt;</pre>	<p><b>The Book</b></p> <p><i>Author:</i> Young, Michael J  <i>Title:</i> XML step by stem  <i>Imprint:</i> Redmond, WA : Microsoft Pr.  <i>Year:</i> 2000</p> <p><i>Author:</i> Williamson, Heather  <i>Title:</i> XML: the complete reference  <i>Imprint:</i> New York : Osborne/McGraw-Hill  <i>Year:</i> 2001</p>
---	---