1. XML 語言

1.1. XML 簡介

XML (*Extensible Markup Language*, 可擴充標記語言) 是由 W3C (*World Wide Web Consortium*, WWW 協會) 所制訂的一種**標記語言** (*Markup Language*)，XML 的規格書可以在 W3C 的網站中下載。

http://www.w3.org/TR/REC-xml

就語言的位階來說，HTML 是 SGML (*Standard Generalized Markup Language*, 標準通用標記語言) 的一種應用；而 XML 是 SGML 的部份集合。一份 XML 文件也是一份 SGML 文件

使用 HTML 的主要目的是在網路上展示文件，而使用 XML 的主要目的是標記資料，以利電腦自動處理；在語言的能力上，HTML 僅能使用預先訂好的元素來標記資料，而 XML 可以自訂所需要的元素來標記資料。

由於 XML 可以自訂元素，所以 XML 可以定義我們所需要元素來描述資料，這個優點使得 XML 比起純文字文件更容易在資料交換中增加新欄位。舉例來說：某兩家公司使用純文字來交換雙方的廠商資料，在廠商資料中擁有三個重要欄位，分別是 “公司名稱”、 “電話_1”、 “電話_2”，就像是下面的例子一樣。

```
Hot Wood Company   9228199928382821
NorthStar Company  292929885858448373
```

假設今天在 “電話_1”、與 “電話_2” 欄位前新增一欄位 “傳真電話” 則雙方交換的文件內容將會變動如下：

```
Hot Wood Company   23919222922819992838282821
NorthStar Company  99493292292929885858448373
```

由上面的例子我們發現，由於資料沒有經過標記處理，所以資料的可讀性很差；另外，已經被修改過的文件如果被舊版處理程式讀取時，將會誤把 “傳真電話” 讀入存成 “電話_1”。

把上述的資料經由標記後的結果爲：

```
<廠商基本資料>
    <廠商>
        <公司名稱>Hot Wood Company</公司名稱>
```

```
        <電話_1>922819992</電話_1>
        <電話_2>838282821</電話_2>
    </廠商>
    <廠商>
        <公司名稱>NorthStar Company</公司名稱>
        <電話_1>292929885</電話_1>
        <電話_2>858448373</電話_2>
    </廠商>
</廠商基本資料>
```

接下來把傳真的欄位加入標記的資料中。

```
<廠商基本資料>
    <廠商>
        <公司名稱>Hot Wood Company</公司名稱>
        <傳真電話>23919222</傳真電話>
        <電話_1>922819992</電話_1>
        <電話_2>838282821</電話_2>
    </廠商>
    <廠商>
        <公司名稱>NorthStar Company</公司名稱>
        <傳真電話>99493292</傳真電話>
        <電話_1>292929885</電話_1>
        <電話_2>858448373</電話_2>
    </廠商>
</廠商基本資料>
```

由上圖可見“傳真電話”欄位很清楚的被安插在“公司名稱”與“電話_1”欄位之間。由於資料經過標記，即使舊版處理程式無法處理“傳真電話”欄位，也不會誤把“傳真電話”讀入存成“電話_1”欄位。所以 XML 在資料欄位異動時，對處理程式的影響比較小。

一份純文字的檔案由於缺乏標記，不容易被剖析處理。而一份 XML 文件可以透過通用處理程式判斷文件是否**形式良好**(*well-formed*)，對於文件的內容可以透過 DTD (*Document Type Declaration,* **文件型別宣告**)或是 XML **文件體制** (*XML Schema*) 來確認 XML 文件是否**有效**(*Valid*)。

圖表 1-1 是一份簡單的 XML 文件。一般而言 XML 文件是以 .xml” 做爲檔案的副檔名，並以純文字的方式儲存，這表示我們可以使用各式各樣的文字編輯器。例如在 Microsoft® Windows® 作業系統中的**記事本**(NotePad.exe)或是 UltraEdit®、XML SPY® 等等；另外在 Unix® 作業系統中的 vi, joe 也都可用來編輯 XML 文件。

```
1  <?xml version = “1.0”?>
2
3  <!--Fig. 3.1 : Hello.xml -->
4  <!-- 一份簡單的 xml 文件 -->
5
6  <Messages>
7    <greeting> 嗨！ Hello! </greeting>
8  </Messages>
```

圖表 1-1

一份 XML 文件應該由 XML 宣告做爲文件的開頭，圖表 1-1 中的第 1 行就是此份 XML 文件的宣告，屬性 version 的值爲 “1.0”，這宣告了這一份 XML 文件符合 XML 1.0 版本的規格，XML 工作小組可能在將來陸續新版。

第 2 行與第 5 行都是**空白類字元**(*White Space*)，在 XML 文件中標籤與標籤之間的空白類字元是被忽略的。第 3, 4 行是 XML 文件中的註解，註解可以出現在文件中**標記**(*Markup*)以外的位置，而且註解文字中不能出現 “--” 的雙連字元。

第 6 到 8 行是這份 XML 文件的主要部份。 XML 文件中必須包含至少一個元素，元素都需要使用起始標籤與結束標籤來定義元素的邊界，例如：<Messages> 就是起始標籤，</Messages> 就是結束標籤，起始標籤與終止標籤的標籤名稱必須要一致，只是結束標籤需要在標籤名稱前加上一條反斜線(“/”)。

此外，XML 文件中是區分大小寫的(*case sensitive*)， “message” 和 “Message” 在 XML 中是被視爲不同的。

所有的 XML 文件都必須要有一個**根元素** (Root Element)，在圖表 1-1 的範例中，其根元素爲 <Messages> 。在 <Messages> 的內容中擁有一個**子元素** (Child Element) <greeting>，這個子元素的內容爲文字 “嗨！ Hello!”。

1.2. 使用瀏覽器觀看 XML 文件

一份**形式良好**(*well-formed*)的文件還不算是一份**有效**(*valid*)的文件，在第四章中我們會討論如何確認一份 XML 文件有效。

```
1   <?xml version="1.0"?>
2
3   <!-- BookStore.xml -->
4   <BookStore>
5       <BookShelf>
6           <Book PublishDate="2002-03-22">
7               <name>Good XML Tips</name>
8               <author>James, Carter</author>
9               <price Currency="US">31.4</price>
10          </Book>
11          <Book PublishDate="2001-10-31">
12              <name>C++ Programming Tips</name>
13              <author>Curtis, Chou</author>
14              <price Currency="US">45.0</price>
15          </Book>
16      </BookShelf>
17      <MagazineShelf>
18          <Magaizne Date="2002-03-24" Type="Monthly">
19              <name>Java Developer Managize</name>
20              <publisher>Forth Group</publisher>
21              <price Currency="US">12.22</price>
22          </Magaizne>
23      </MagazineShelf>
24      <NewspaperShelf>
25          <Newspaper Date="2002-05-25">
```

```
26              <name>China Times</name>
27              <heading>
28                   <item>The fastest CPU is born.</item>
29                   <item>Weather is getting warm in this week</item>
30              </heading>
31              <price Currency="NT">15</price>
32          </Newspaper>
33      </NewspaperShelf>
34  </BookStore>
```

圖表 1-2

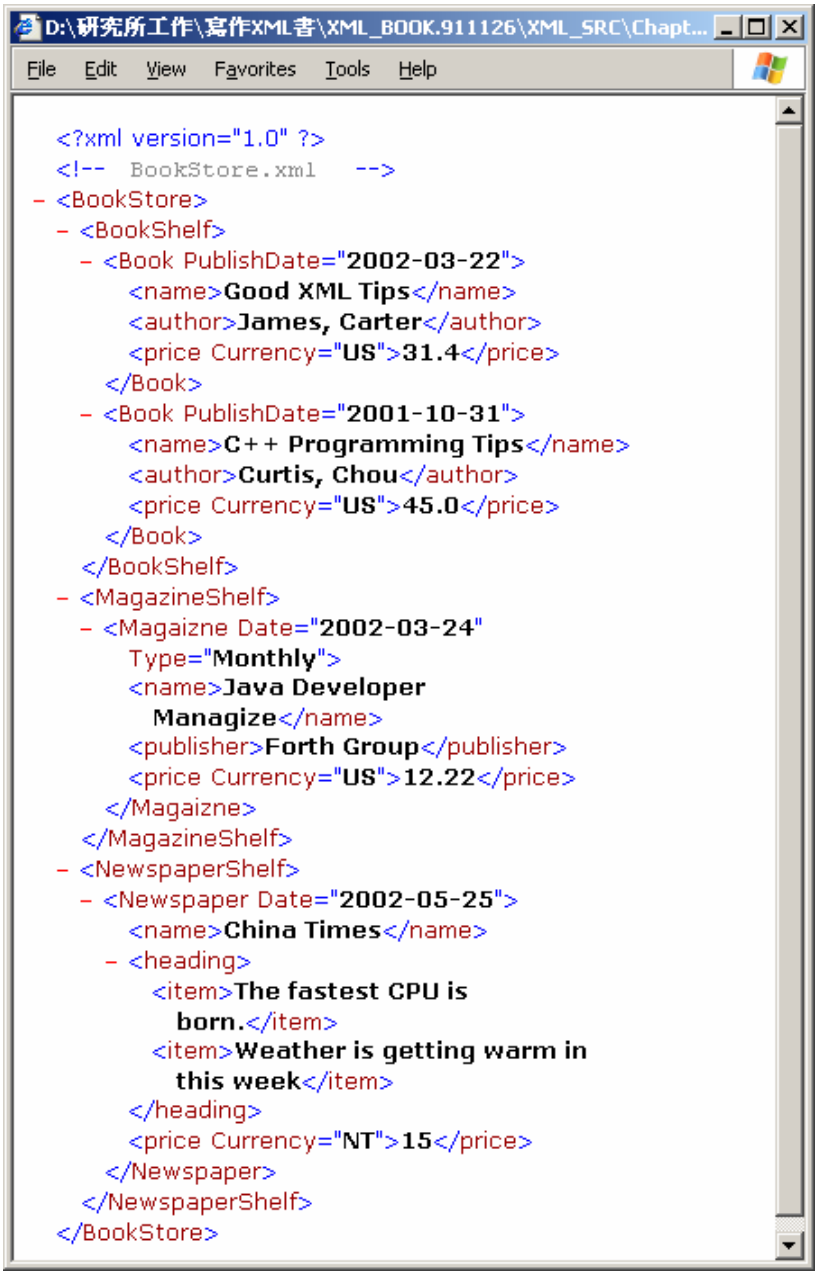
在圖表 1-2 所看到的是一份 XML 文件的原始內容。當使用記事本應用程式觀看時，比較不容易掌握整份 XML 文件的樹狀結構，特別是在這一份 XML 文件很長或是很深的時候。

圖表 1-3 上方的兩個畫面是我們使用 Microsoft® Internet Explorere 6 開啓 BookStore.xml 的畫面，在畫面中我們可以見到 XML 文件中的 XML 宣告與註解，接下來可以看到一個由“＋”與“－”符號組成的樹狀結構文件。當元素前面顯示“＋”表示這個元素擁有**內容**(*content*)，我們可以點選“＋”將其展開。當內容已經被完全展開之後，該元素前方的“＋”號會變成“－”號；反之若元素前面顯示“－”， 表示這個元素的內容已經完全展開，若點選“－”則元素內容將被隱藏。

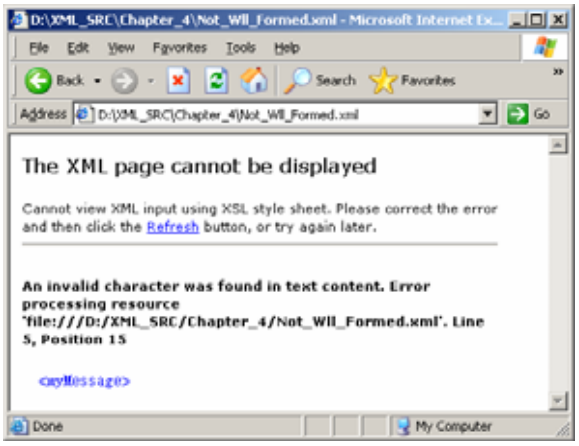
如果我們的 XML 文件含有錯誤，則使用瀏覽器將會有錯誤訊息出現，如圖表 1-4 中下方的畫面。。

目前我們所看到的呈現方式是 Microsoft® Internet Explorere 6 內定的顯示方式。然而，使用者自己也可以訂立自己希望呈現的方式，透過 XSLT (*Extensible Stylesheet Language Transformation*, **可擴充樣式表語言**) 可以將 XML 文件轉換成其他樹狀結構文件，例如 HTML。這將在第六章以後討論。


D:\研究所工作\寫作XML書\XML_B...
File Edit View Favorites Tools
<?xml version="1.0" ?>
+ <BookStore>


D:\研究所工作\寫作XML書\XML_BOOK.911126\XML_SRC\Chapt...
File Edit View Favorites Tools Help
<?xml version="1.0" ?>
<!-- BookStore.xml -->
- <BookStore>
- <BookShelf>
- <Book PublishDate="2002-03-22">
<name>Good XML Tips</name>
<author>James, Carter</author>
<price Currency="US">31.4</price>
</Book>
- <Book PublishDate="2001-10-31">
<name>C++ Programming Tips</name>
<author>Curtis, Chou</author>
<price Currency="US">45.0</price>
</Book>
</BookShelf>
- <MagazineShelf>
- <Magaizne Date="2002-03-24"
Type="Monthly">
<name>Java Developer
Managize</name>
<publisher>Forth Group</publisher>
<price Currency="US">12.22</price>
</Magaizne>
</MagazineShelf>
- <NewspaperShelf>
- <Newspaper Date="2002-05-25">
<name>China Times</name>
- <heading>
<item>The fastest CPU is
born.</item>
<item>Weather is getting warm in
this week</item>
</heading>
<price Currency="NT">15</price>
</Newspaper>
</NewspaperShelf>
</BookStore>

圖表 1-3



圖表 1-4

1.2.1. 文字與標記

依據 XML 規格書的定義，一份 XML 文件是由文字與標記夾雜構成。

1.2.2. 標記

依據 XML 規格書，標記包含**起始標籤**(*start-tag*)、**終止標籤** (*end-tag*)、**空白元素標籤** (*empty-element tag*)、**實體參用** (*entity reference*)、**字元參用** (*character reference*)、**CDATA 段分隔符號** (*CDATA section delimiters*)、**文件型別宣告** (*document type delcarations*,DTD)、**處理指令** (*processing instructions*,PI)、**XML 宣告** (*XML declarations*)、**文字宣告** (*text declarations*) 等。

1.2.3. 字元集

XML 文件所使用的是 *Unicode* ，在 *Unicode* 中內含了 ASCII(*American Standard Code for Information Interchange*) 以及其他語言，例如中文、日文等，詳細的 Unicode 資訊可參閱：

http://www.unicode.org

有一些字元由於不方便輸入，例如阿拉伯文，可以使用 **字元參用**(*character*

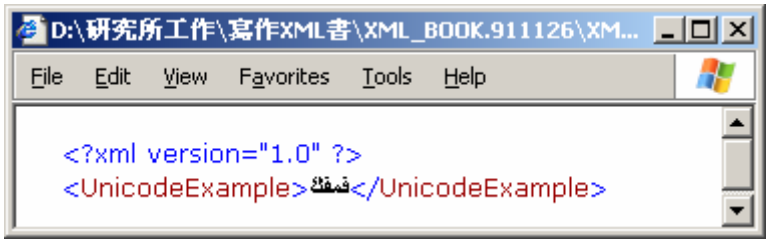
reference) 來解決。字元參用的形式是由 '&#' 開始，中間接續該 Unicode 字元的代碼，最後由 ';' 結束。

下圖表 1-5 是示範了在 XML 文件中使用字元參用顯示四個阿拉伯文字元。

```
1  <?xml version="1.0" ?>
2  <UnicodeExample>
3     &#1602;&#1605;&#1602;&#1603;
4  </UnicodeExample>
```

圖表 1-5

圖表 1-5 在 Microsoft® Internet Explorere 6 中顯示的畫面：



圖表 1-6

幾乎所有的字元都可以被用來寫作 XML 文件，不過仍有一些字元是被 XML 所保留的，而且保留字是不可以出現在**內容**(*content*)之中的。

被保留的字元分別是：

'&' , '<'

這是因爲這幾個符號是用來表示 XML 標記的重要符號，所以不可以出現在內容中。但爲了使用這些保留字，所以必須改用**實體參用**(*Entity Reference*)來表示該符號；實體參用由 '&' 開始，接續一個實體參用名稱，最後由一個 ';' 作結束。

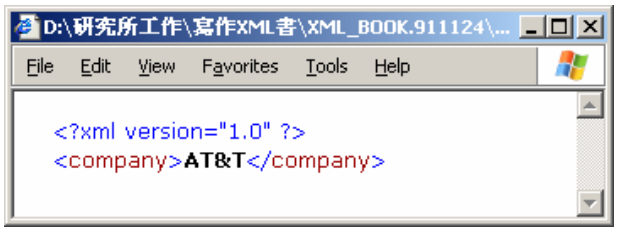
假設我們想要在內容中表示

<company> AT&T </company >

由於 '&' 是一個保留字，所以我們必須將上面的表示式改寫成：

<company> AT&T </company>

一旦 & 被讀取時，就會置換成 '&'，如下圖 3.7 所示。



圖表 1-7

諸如此類的保留字在內容中都必須改用實體參用，上述的幾個符號都是屬於 **內建實體**(*build-in entities*) ，當然我們也可以自訂實體，這將在後續章節討論。

下面表格列出了幾個常用的內建實體。

符號	實體參用
&	&amp;
‘	&apos;
”	&gt;
>	&lt;
<	&quot;

空白類字元(*white space*)包含**空白字元**(*space*)、**定位點字元**(*tab*)、**換行字元**（*line feed*）、**歸位字元**(*carriage return*) 等，這一些空白類字元可能因爲處理程式的不同，可能被視爲顯著的 (Explicit) 或是不顯著 (Implicit)。

也就是說，多的空白類字元可能被應用程式縮減或是保留下來。

例如：

```
<greeting> Hi!          I am here.</greeting>
```

將可能被應用程式縮減成爲：

```
<greeting> Hi! I am here.</greeting>
```

1.2.4. 元素

根據 XML 規格書的定義，**元素**(*element*)是由**起始標籤**(*start-tag*)、**內容**(*content*)、**終止標籤**(*end-tag*)所構成。

起始標籤是由‘<’開始，接續一個標籤名稱，在標籤名稱後面是一些可有可無的屬性，最後由‘>’結束。例如：

```
<Book PublishDate=”2002-03-22” Catalog=”CS”>
```

在上面的例子中， “Book” 是該標籤的名稱，而 **PublishDate=”2002-03-22”** 和 **Catalog=”CS”** 是 “Book” 元素的屬性；屬性的定義是由**屬性名稱**、**等號**、**屬性值**所構成。

終止標籤是由 **’</’** 開始，並接續一個標籤名稱，最後由**’>’** 結束。所以相對於上例的起始標籤，其終止標籤如下：

```
</Book>
```

在標籤內的內容可以包括**字元資料**(*character data*)、**元素**(*element*)、**參用**(*reference*)、**CDATA 段** (*CDATA section*)、**處理指令**(*processing instruction, PI*)、**註解**(*comment*)構成。例如：

```
<Book PublishDate="2002-03-22">
   <!-- Popular Book -->
   <name>Good XML Tips</name>
   <author>James, Carter</author>
   <price Currency="US">31.4</price>
</Book>
```

除此之外，元素可以是沒有內容的，這種元素被稱爲**空白元素**(*empty element*)。

例如在 HTML 中的 <img> 元素就是空白元素，而空白元素寫作的形式有兩種，一種是擁有起始標籤與傳統標籤但沒有內容的形式，例如：

```
<img src =”Duke.jpg”></img>
```

空白元素的始標籤與終止標籤可以合併，例如上一行可以被合併成爲：

```
<img src =”Duke.jpg”/>
```

1.2.5. 註解

依據 XML 規格書的定義，**註解**(*comment*)是由‘<!--’開始接續註解文字，最後結束於‘-->’，而在註解文字中不可以出現雙連減號‘--’，註解文中的最後一個字元也不可以是減號‘-’。

下列的例子是一個合法的註解：

```
<!-- 在根元素 <html> 下的子元素僅有 <head> & <body> -->
```

接下來示範的例子皆是不合法的註解。

```
<!-- XSL--Transformation -->
```

上例不合法是因爲在註解文字中出現了雙連減號‘--’。

```
<!-- 不合法的註解 --->
```

上例不合法是因爲在註解文字中的最後一個字元爲‘-’。

1.2.6. 處理指令

在 XML 文件中允許包含 **處理指令**(*Processing Instructions*, PI)。處理指令是由‘<?’開始接續處理指令，最後由‘?>’結束。而處理指令中不可以出現“XML”爲名稱的處理指令，而且 ″XML″ 三個字元是不區分大小寫的。不可以使用 ″XML″ 作爲處理指令名稱的原因是該名稱已經用於 XML 文件的宣告中了。

處理指令是由應用程式依據需求自行定義。而下例是一個處理指令，該指令告知應用程式 XmlSpy 讀取特定的檔案 datasheet.sps。

```
<?xmlspysps datasheet.sps?>
```

1.2.7. CDATA 段

XML 文件中，希望在寫作內容時（例如記錄一段 C＋＋或是 JAVA 的程式碼），而不受到保留字的規範，這時我們就需要用到 **CDATA 段**(*CDATA section*)。

在圖表 1-8 中第 4 到 12 行之間是描述一段 C＋＋程式碼，這段程式碼是 <code> 元素的內容。因爲沒有使用 CDATA 區塊，所以在寫作 C＋＋ 範例程式時，必須小心謹慎的避開所有保留字元，所以短短幾行出現在 XML 文件中的 C＋＋範例程式碼，需要花費許多時間來將保留字換成實體參用。

而第 13 到 22 行之間的內容使用了 CDATA 段，所以不需要將保留字替換成實體參用；CDATA 段是由 ‘**<![CDATA[**’開頭，並由‘**]]>**’結束。

```
1   <?xml version="1.0"?>
2   <!—Fig 3.10 : Codes.xml -->
3   <codes>
4       <code>
5           <!--不使用 CData 區塊的作法 -->
6           int* ptr_i;
7           int i=20;
8           ptr_i=&amp;i;
9           if (*ptr_i &gt; 100 ) {
10              cout &lt;&lt;&quot; i is &gt; 100. &quot;;
11          }
12      </code>
13      <code>
14          <![CDATA[
```

```
15              int* ptr_i;
16              int i=20;
17              ptr_i=&i;
18              if (*ptr_i > 100 ) {
19                   cout <<" i is > 100. ";
20              }
21          ]]>
22      </code>
23  </codes>
```

圖表 1-8

但是 CDATA 區塊中仍存在一個保留字，那就是 ‘]]>’，下面的例子是一個錯誤示範。

```
<![CDATA[
        這是一個錯誤示範！ ]]>
]]>
```

上面的例子就是因爲多了一個 ‘]]>’，導致整個 CDATA 區塊找不到成對的標籤，因此是一個不合法的 CDATA 段。

1.3. 形式良好的 XML 文件

依據 XML 規格書的定義，一份形式良好的 XML 文件必須要符合下列三項規定。

一、XML 文件必須合乎規格書定義的文法。
二、必須符合規格書中所有對形式良好的限制。
三、每一個在文件內的**已剖析實體**(*parsed entity*) 不論直接或是間接被**參用**(*reference*)均需是**形式良好**(*well-formed*)的。

以下是 XML 規格書中幾個重要的**形式良好性限制**(*well-formed constraint*)。

・**起始標籤**(*start-tag*)的**名稱**(*name*)必須和**結束標籤**(*end-tag*)一致。
・在**起始標籤**(*start-tag*)或是**空白元素標籤**(*empty-element tag*)不可出現相同的屬性名稱。
・屬性值不可以包含對**外部實體**(external entity)直接或是間接的**實體參用**(*entity reference*)。
・文件中的合法字元必須是 *Unicode* 字元集中的第 #x9, #xA, #xD, [#x20-#xD7FF], [#xE000-#xFFFD], [#x10000-#x10FFFF]

```
1  <?xml version = “1.0”?>
2  <!-- Fig. 3.3:文件中沒有任何元素 -->
```

圖表 1-9

圖表 1-9 並非形式良好的文件，因爲文件中沒有任何元素。依據規格書中的 XML 文法，所有 XML 文件中都必須包含至少一個元素。

```
1  <?xml version="1.0" ?>
2
3  <Messages>
4      <message>A1</message>
5      <message>A2</message>
6  </Messages>
7  <Messages>
8      <message>A3</message>
9  </Messages>
```

圖表 1-10

圖表 1-10 也不是一份形式良好的 XML 文件，因爲這個範例中出現了兩個根元素 <Messages>。根據 XML 文法，一份形式良好的 XML 文件只能擁有一個根元素。

```
1  <?xml version = “1.0”?>
2
3  <!—Fig. 3.5 : Not_Well_Formed.xml -->
4  <Messages>
5    <greeting> 第一個根元素 </greeting>
6  </Messages>
7  <Items> 第二個根元素
8    <myFirstItem>
9      <mySecondItem>
10   </myFirstItem>
11     </mySecondItem>
12 </Items>
```

圖表 1-11

圖表 1-11 也非形式良好的文件，除了擁有兩個根元素 <Messages>, <Items> 之外，第 8 到 11 行的標籤沒有依照巢狀的次序排列。不同組的起始元素和終止元素禁止交錯重疊在一起。

```
1  <?xml version="1.0" ?>
2
3  <Hello>
4      Good morning.
5  </Hi>
```

圖表 1-12

圖表 1-12 不是形式良好的 XML 文件，因爲根元素的**起始標籤**(*start-tag*)

名稱與**結束標籤**(*end-tag*)名稱沒有一致，所以沒有合乎**形式良好性限制**(*well-formed constraint*)的規定。

1.4. XML 命名空間

XML 允許使用者自行建立標籤，這使得不同的 XML 文件創作者可能創造出相同名稱但定義不同的元素標籤，如此的情況就稱爲**命名衝突**(naming collisions)。爲了解決命名衝突，W3C 制訂了**命名空間**(*namespace*)，利用 URI (*Uniform Resource Identifier*, **一致性資源辨識符號**) 辨識來自不同出處的元素或是屬性，關於命名空間的規格書可到 W3C 網站下載。

http://www.w3.org/TR/REC-xml-names/

假設某甲使用 <name> 作爲他記錄足球明星的元素標籤；而某乙使用了 <name> 作爲記錄他所喜愛的棒球明星的元素標籤。如果將甲乙兩人的 XML 文件合併時，那麼 <name> 到底是要以甲方還是乙方的定義？

此時我們就需要利用 XML 命名空間的機制來解決這個問題。首先我們在標籤名稱前面冠上命名空間名稱再加上一個冒號 '：' 隨後跟著標籤的名稱，就像是這個樣子：

```
<甲:name>John Wu</甲:name>
<乙:name>Ronald Do</乙:name>
<甲:name>Peter Wang</甲:name>
```

所以上面的 '甲:' 和 '乙:' 是**命名空間前置名稱**(Namespace Prefix)，而且每一個前置名稱要與 URI 相結合在一起；URI 可以是一個 URL 也可以是一個檔案位置，這個 URI 不一定要真實存在，只要同一份文件中不要出現同樣的 URI 即可，但是要符合 URI 的寫作規定。

關於 URI 的寫作規定可查閱：

http://www.ietf.org/rfc/rfc2396.txt

前置名稱與對應的 URI 宣告必須位於 XML 文件中的根元素中。宣告的方式爲

```
'xmlns:'+ 命名空間前置名稱 + '=' + URI
```

例如甲、乙、丙三個命名空間前置名稱在根元素中的宣告如下：

```
<根元素 xmlns:甲="http://www.甲.org"
        xmlns:乙="http://www.2nd.edu.tw"
        xmlns:丙="http://www.third.com">
    <!-- 略 -->
</根元素>
```

下列圖表 1-13 是命名空間的一個應用實例：

```
1   <?xml version="1.0" ?>
2   <!--Fig. 3.11 : Programmer.xml -->
3   <programmer xmlns:java="http://www.java.dep.com"
4               xmlns:c="http://www.prog.c.lang.edu"
5               xmlns:vb="http://www.mscode.vbclub">
6       <java:name>Mahler Chou</java:name>
7       <c:name>Andy Shi</c:name>
8       <c:name>Mary Tai</c:name>
9       <vb:name>Puccini W. Hong</vb:name>
10  </programmer>
```

圖表 1-13

在圖表 1-13 中，共宣告了三個不同的命名空間前置符號，分別爲 java, c, vb。

```
<programmer
  xmlns:java = "http://www.java.dep.com"
  xmlns:c    = "http://www.prog.c.lang.edu"
  xmlns:vb   =  "http://www.mscode.vbclub">
```

這三個命名空間前置符號都有相對應的 URI，用於區隔第 6 到 9 行的 <name> 元素。

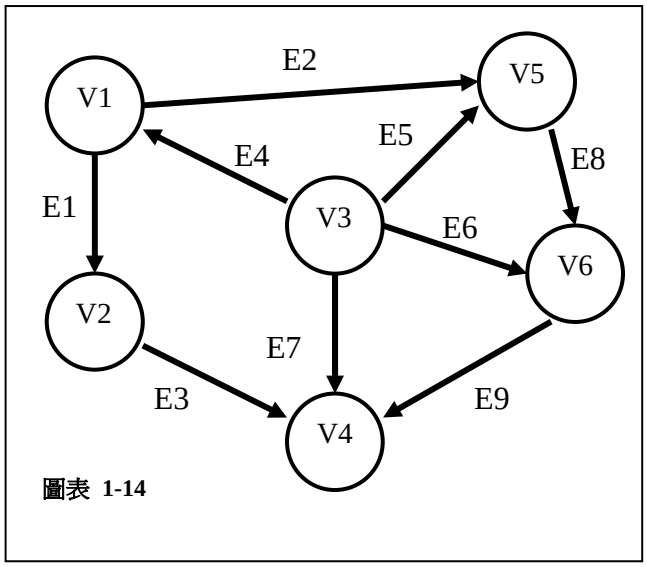
1.5. 個案研究

在本章討論完 XML 語言之後，相信我們對 XML 文件寫作上有了初步概念，爲了加深印象，最後使用兩個例子來示範 XML 文件的寫作。

1.5.1. 使用 XML 描述有向圖形

下列圖表 1-14 是一個由六個**頂點**(Vertex) 與九條**邊**(Edge)所構成的**有向圖形** (Directed Graph)。爲了描述該有向圖形，我們分析了圖表 1-14 得知每一個頂點與邊都自己獨一無二的名字，而邊還要記錄出發和目的的頂點名稱。所以我們設計了一份描述有向圖形的 XML 文件由 <digraph> 作爲根元素，在 <digraph> 中可以出現任意多個 <vertice> 或是 <edge> ，其中 <vertice> 和 <edge> 都有

一個 name 屬性用來記錄頂點名稱;而 <edge> 另外還有 from 屬性記錄由那個頂點發出，而 to 屬性記錄目的頂點爲何。



圖表 1-14

下圖是圖表 1-14 中的有向圖形之 XML 描述檔如下：

```
1   <?xml version="1.0"?>
2   <!-- Fig. 3.13 : digraph.xml -->
3   <!-- 有向圖形 XML 描述檔案 -->
4   <digraph>
5       <vertice name="V1"/>
6       <vertice name="V2"/>
7       <vertice name="V3"/>
8       <vertice name="V4"/>
9       <vertice name="V5”/>
10      <vertice name="V6"/>
11      <edge name="E1" from="V1" to="V2"/>
12      <edge name="E2" from="V1" to="V5"/>
13      <edge name="E3" from="V2" to="V4"/>
14      <edge name="E4" from="V3" to="V1"/>
15      <edge name="E5" from="V3" to="V5"/>
16      <edge name="E6" from="V3" to="V6"/>
17      <edge name="E7" from="V3" to="V4"/>
18      <edge name="E8" from="V5" to="V6"/>
19      <edge name="E9" from="V6" to="V4"/>
20  </digraph>
```

圖表 1-15

1.5.2 使用 XML 記錄學生基本資料與成績

某大學資管系將系上學生基本資料與成績以 XML 文件儲存，以利日後由

電腦自動處理，以節省學校資源與提高工作效率。

然而該系目前擁有大學部與碩士班學生若干名；大學部的班級原本應該有一至四年級，可是由於該系僅成立兩年，所以目前僅有大一與大二學生而已，爲了日後作業方便，在分類學生時，仍然需要預留大三與大四的欄位；在大學部中的學生資料請依據年級、班級整理，而學生資料中必須包含學號與姓名，另外還能夠記錄不同考試的科目、日期與成績。

而碩士班的學生資料室依據入學年度整理入檔；在碩士班的學生資料中只比大學的學生資料多了一個記錄學生入學的身份爲“一般生”或是“在職生”。

以下是資管系所提供的基本資料：

資管系班級明細	
班級/班別	**導師**
資管系大學部一年級/A	黃德高
資管系大學部一年級/B	陳望重
資管系大學部二年級/A	楊清廉
資管系碩士班/91 年	(不設導師)

資管系學生明細		
班別	**學號**	**姓名**
資管系大學部一年級/A	2382822	陳大雄
資管系大學部一年級/A	2388952	胡宜靜
資管系大學部一年級/B	2384823	林光明
資管系大學部一年級/B	2348953	王正大
資管系大學部二年級/A	1386822	陳小花
資管系大學部二年級/A	1386952	黃望月
資管系大學部二年級/A	1386939	王強
資管系碩士班/91 年	91382399	林叮噹
資管系碩士班/91 年	91386939	李丸子
資管系碩士班/91 年	91386453	陳一光
資管系碩士班/91 年	91386666	李四

資管系期中考成績明細			
科目	**考試日期**	**學號**	**成績**
資料處理	2002-03-21	2382822	83
資料處理	2002-03-21	2388952	88
資料處理	2002-03-21	2384823	95
資料處理	2002-03-21	2348953	99
資訊概論	2002-03-21	2382822	54
資訊概論	2002-03-21	2388952	76
資訊概論	2002-03-21	2384823	57

資訊概論	2002-03-21	2348953	96
管理學	2002-03-22	2382822	66
管理學	2002-03-22	2388952	74
管理學	2002-03-22	2384823	89
管理學	2002-03-22	2348953	94
C 語言程式設計	2002-03-20	1386822	24
C 語言程式設計	2002-03-20	1386952	34
C 語言程式設計	2002-03-20	1386939	84
資訊概論	2002-03-23	1386822	54
資訊概論	2002-03-23	1386952	94
資訊概論	2002-03-23	1386939	34
統計學	2002-03-24	1386822	66
會計學	2002-03-24	1386939	49
軟體技術	2002-03-22	91382399	88
編譯理論	2002-03-22	91386666	86

下列圖表 1-16 是綜合了上述文件所整理出來的學生基本資料與成績的 XML 應用範例：

```
<?xml version="1.0"?>
<資管系>
    <大學部>
        <一年級>
            <班級 班別="A" 導師="黃德高">
                <學生>
                    <學號>2382822</學號>
                    <姓名>陳大雄</姓名>
                    <成績 類別="期中考">
                        <科目 日期="2002-03-21" 名稱="資料處理">83</科目>
                        <科目 日期="2002-03-21" 名稱="資訊概論">54</科目>
                        <科目 日期="2002-03-22" 名稱="管理學">66</科目>
                    </成績>
                </學生>
                <學生>
                    <學號>2388952</學號>
                    <姓名>胡宜靜</姓名>
                    <成績 類別="期中考">
                        <科目 日期="2002-03-21" 名稱="資料處理">88</科目>
                        <科目 日期="2002-03-21" 名稱="資訊概論">76</科目>
                        <科目 日期="2002-03-22" 名稱="管理學">74</科目>
                    </成績>
                </學生>
            </班級>
            <班級 班別="B" 導師="陳望重">
                <學生>
                    <學號>2384823</學號>
                    <姓名>林光明</姓名>
                    <成績 類別="期中考">
                        <科目 日期="2002-03-21" 名稱="資料處理">95</科目>
                        <科目 日期="2002-03-21" 名稱="資訊概論">57</科目>
                        <科目 日期="2002-03-22" 名稱="管理學">89</科目>
                    </成績>
                </學生>
```

```
                    <學生>
                        <學號>2348953</學號>
                        <姓名>王正大</姓名>
                        <成績 類別="期中考">
                            <科目 日期="2002-03-21" 名稱="資料處理">99</科目>
                            <科目 日期="2002-03-21" 名稱="資訊概論">96</科目>
                            <科目 日期="2002-03-22" 名稱="管理學">94</科目>
                        </成績>
                    </學生>
                </班級>
            </一年級>
            <二年級>
                <班級 班別="A" 導師="楊清廉">
                    <學生>
                        <學號>1386822</學號>
                        <姓名>陳小花</姓名>
                        <成績 類別="期中考">
                            <科目 日期="2002-03-20" 名稱="C 語言程式設計">24</科目>
                            <科目 日期="2002-03-23" 名稱="資訊概論">54</科目>
                            <科目 日期="2002-03-24" 名稱="統計學">66</科目>
                        </成績>
                    </學生>
                    <學生>
                        <學號>1386952</學號>
                        <姓名>黃望月</姓名>
                        <成績 類別="期中考">
                            <科目 日期="2002-03-20" 名稱="C 語言程式設計">34</科目>
                            <科目 日期="2002-03-23" 名稱="資訊概論">94</科目>
                        </成績>
                    </學生>
                    <學生>
                        <學號>1386939</學號>
                        <姓名>王強</姓名>
                        <成績 類別="期中考">
                            <科目 日期="2002-03-20" 名稱="C 語言程式設計">84</科目>
                            <科目 日期="2002-03-23" 名稱="資訊概論">34</科目>
                            <科目 日期="2002-03-24" 名稱="會計學">49</科目>
                        </成績>
                    </學生>
                </班級>
            </二年級>
            <三年級/>
            <四年級/>
        </大學部>
        <碩士班>
            <班級 入學年度="91">
                <學生 類型="一般生">
                    <學號>91382399</學號>
                    <姓名>林叮噹</姓名>
                    <成績 類別="期中考">
                        <科目 日期="2002-03-22" 名稱="軟體技術">88</科目>
                    </成績>
                </學生>
                <學生 類型="一般生">
                    <學號>91386939</學號>
                    <姓名>李丸子</姓名>
                </學生>
                <學生 類型="一般生">
                    <學號>91386453</學號>
                    <姓名>陳一光</姓名>
                </學生>
                <學生 類型="在職生">
                    <學號>91386666</學號>
```

```
98                              <姓名>李四</姓名>
99                              <成績 類別="期中考">
100                                 <科目 日期="2002-03-22"  名稱="編譯理論">86</科目>
101                             </成績>
102                         </學生>
103                 </班級>
104         </碩士班>
105 </資管系>
```

圖表 1-16

1.6. 網際網路資源

http://www.w3.org/XML
W3C 官方網站，在這裡我們可以找到 XML 的規格書，還有其他關於 XML 技術的相關資源。

http://www.xml.com
有相當豐富資源的 XML 網站，在網站中可以找到文章、新聞等等。

http://www.xml.org
XML 入口網站，在這裡可以找到許多相關的 XML 網站連結。