

第 7 章 可擴充樣式格式化物件 語言(XSL-FO)

7.1 XSL-FO 簡介

XSL 語言是由兩個部份組成，分別是前面章節所介紹的 XSLT 與本章節所要介紹的 XSL-FO。XSL-FO 是 Extensible Stylesheet Language Formatting Objects 的縮寫。XSL-FO 是用來描述一份文件的輸出，透過 XSL-FO 所提供的元素來定義紙張大小、書寫方向、邊界、內文、段落、字體大小、顏色等等資訊後，再交由 XSL-FO 處理器來產生最終的輸出，例如：PDF 檔案、Microsoft Word 格式檔案、Rich Text File 格式檔案、純文字檔案等。

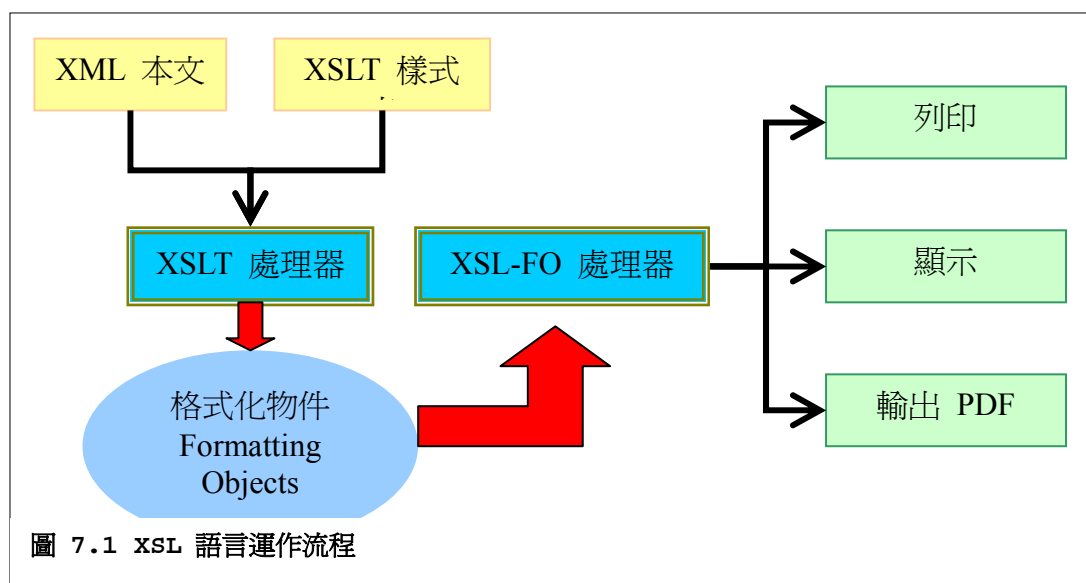
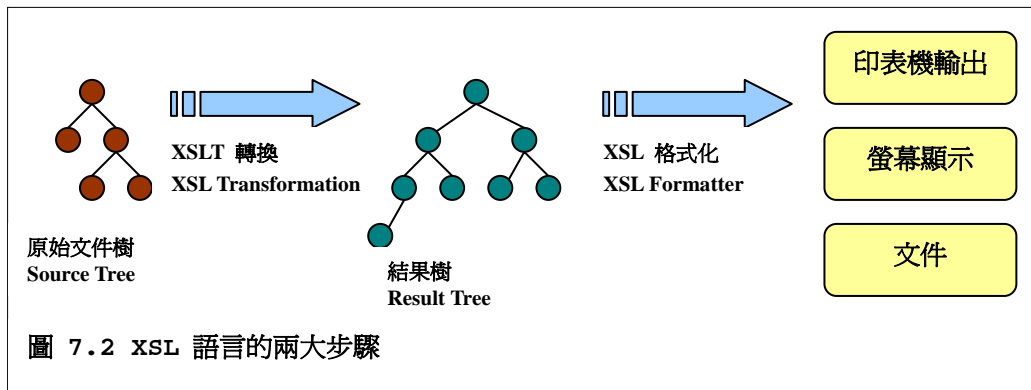


圖 7.1 是 XSL 在應用上的流程。首先是使用 XSLT 將 XML 文件轉換成 XSL-FO 語言，然後再將產生出的 XSL-FO 語言交由特定的 XSL-FO 處理器(例如輸出的目的為 PDF 格式檔案的 XSL-FO 處理器)。所以透過 XSL 可以將 XML 文件排版成為制式的文件輸出，這一個優點解決了 HTML 文件在列印時造成的分頁或是排版上的問題。



由於 XSL-FO 語言相當的複雜，這一點光由規格書高達四百餘頁可以見得。雖然 XSL-FO 本身很複雜，但是我們應用的部份通常都是很規則的（例如定義紙張大小、段落等），所以使用者僅需要瞭解如何設計 XSLT 樣式表，透過樣式表將原始文件樹轉化成 XSL-FO 結果樹（圖 7.2），所需要修改排版的規則，僅需要修改 XSLT 樣式表即可。

在本章中會先以手寫的方式介紹如何使用 XSL-FO 定義一份文件，並將之產生成 PDF 格式的檔案，並使用 Adobe Acrobe® Reader 來檢視結果，章節的最後再教導如何設計 XSLT 樣式表，並透過 XSLT 來產生 XSL-FO，就和圖 7.1 一般。

7.2 XSL-FO 處理器

XSL-FO 的處理器可於 W3C 網站(<http://www.w3.org>)上下載，比較著名而且是免費的 XSL-FO 處理是由 Apache 所發展的 FOP (Formatting Objects Processor)，這套軟體是以 Java 語言所開發並可以輸出產生 PDF、PCL（惠普所發展的印表機控制語言）、PostScript、RTF、SVG、AWT、MIF、純文字等等的檔案。

要使用 FOP 首先先至 Apache 網站下載新版的 FOP 軟體。

FOP 所在網址如下：

<http://xml.apache.org/fop/index.html>

FOP 下載位置：

<http://xml.apache.org/dist/fop/#>

在 FOP 的下載位置中有許多檔案（圖 7.3），我們可以下載檔名為 fop-0.20.4-bin.tar.gz 的檔案。在下載位置中的檔案均會以 fop 作為開頭，然後緊跟著版本的編號，最後會冠上 bin 或是 src。bin 和 src 不同點在於 src 多了 FOP 的原始碼，對於有心研究 FOP 運作的人，可以改下載 fop-0.20.4-src.tar.gz。

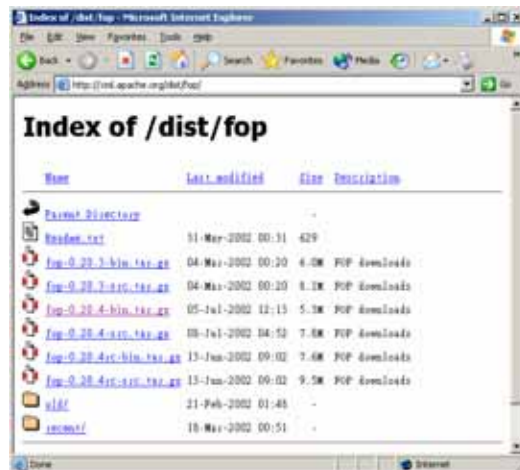


圖 7.3 FOP 下載位置

7.2.1 安裝 FOP

檔 fop-0.20.4-bin.tar.gz 下載完成後，請使用 Winzip 或是 WinRAR 將檔案解開，解開後會得到 fop-0.20-4-bin.tar。我們必須再透過 Winzip 或是 WinRAR 再將檔案解開一次，但是這一次將解壓縮的目的地改至磁碟機的根本目錄下，例如 C:\或是 D:\。

當解壓縮完畢之後，在 FOP 的目錄中應該會有圖 7.4 相同的檔案與資料夾。

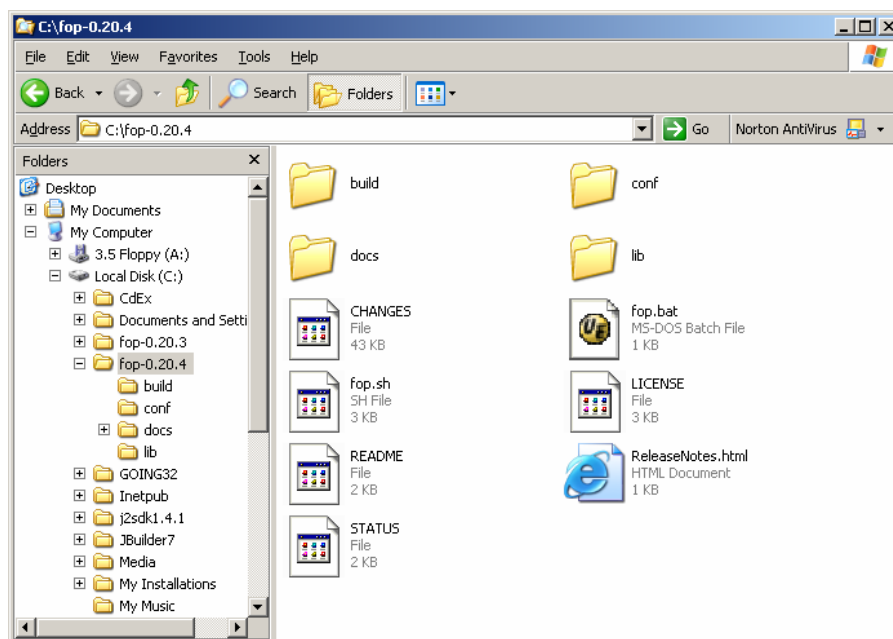


圖 7.4 FOP 資料夾內容

由於 FOP 是以 JAVA 語言所開發，所以要執行 FOP 必須要有 Java 1.2.x 以上的執行環境才能運作。您可以到 Sun 網站 (<http://java.sun.com>) 下載最新的 Java RuntimeEnvironment 或是 JDK 安裝至您的電腦後方可使用 FOP。

7.2.2 如何使用 FOP

要執行 FOP 必須先開啓命令列模式，並設定好 Java 環境變數，然後至 FOP 目錄中執行 Fop.bat。Fop 執行的參數如下：

```
Fop [選項] [-fo | -xml] 輸入檔案 [-xsl 檔案]
      [-awt | -pdf | -mif | -pcl | -txt | -svg | -at | -print] <輸出
      檔案>
```

Fop 選項	
-d	Debug 模式
-x	傾印設定
-q	安靜模式
-c cfg.xml	使用額外的設定檔 cfg.xml
-l lang	設定要使用的語言

圖 7.5 Fop 選項參數

Fop 輸入檔案	
輸入檔案名稱	xsl:fo 輸入檔案 (與下一個相同)
-fo 檔案名稱	xsl:fo 輸入檔案
-xml 檔案名稱	xml 檔案，這個必須要和 -xsl 並用
-xsl 樣式表檔名	xslt 樣式表

圖 7.6 Fop 輸入的檔案

Fop 輸出檔案類型	
輸出檔案	預設輸出檔案類型為 PDF 格式檔案
-pdf 輸出檔名	以 PDF 檔案格式輸出結果
-awt	顯示在螢幕上
-mif 輸出檔名	以 MIF 檔案格式輸出
-pcl 輸出檔名	以 PCL 檔案格式輸出
-txt 輸出檔名	以純文字檔案格式輸出
-svg 輸出檔名	以 SVG 分頁檔案格式輸出
-at 輸出檔名	以 XML 形式輸出 Area Tree
-print	執行輸出至印表機(詳見 -print help)

圖 7.7 Fop 輸出的檔案類型

Fop 操作範例	
Fop foo.fo foo.pdf	將 foo.fo 輸出成爲 foo.pdf
Fop -fo foo.fo -pdf foo.pdf	(和上一個範例相同)
Fop -xsl foo.xsl -xml foo.xml -pdf foo.pdf	分別指定樣式表與 XML 文件，並輸出 PDF 檔案。
Fop foo.fo - mif foo.mif	輸出 mif 類型檔案
Fop foo.fo -print	輸出至印表機
Fop foo.fo -awt	輸出至螢幕

圖 7.8 Fop 操作範例

在命令列參數下執行的結果：

```

C:\fop-0.20.4>fop MyFo.fo MyFo.pdf

C:\fop-0.20.4>java -cp build\fop.jar;lib\batik.jar;lib\xalan-2.3.1.jar;lib\xercesImpl-2.0.1.jar;lib\xml-apis.jar;lib\avalon-framework-cvs-20020315.jar;lib\logkit-1.0.jar;lib\jimi-1.0.jar org.apache.fop.apps.Fop MyFo.fo MyFo.pdf
[INFO] FOP 0.20.4
[INFO] building formatting object tree
[INFO] [1]
[INFO] Parsing of document complete, stopping renderer

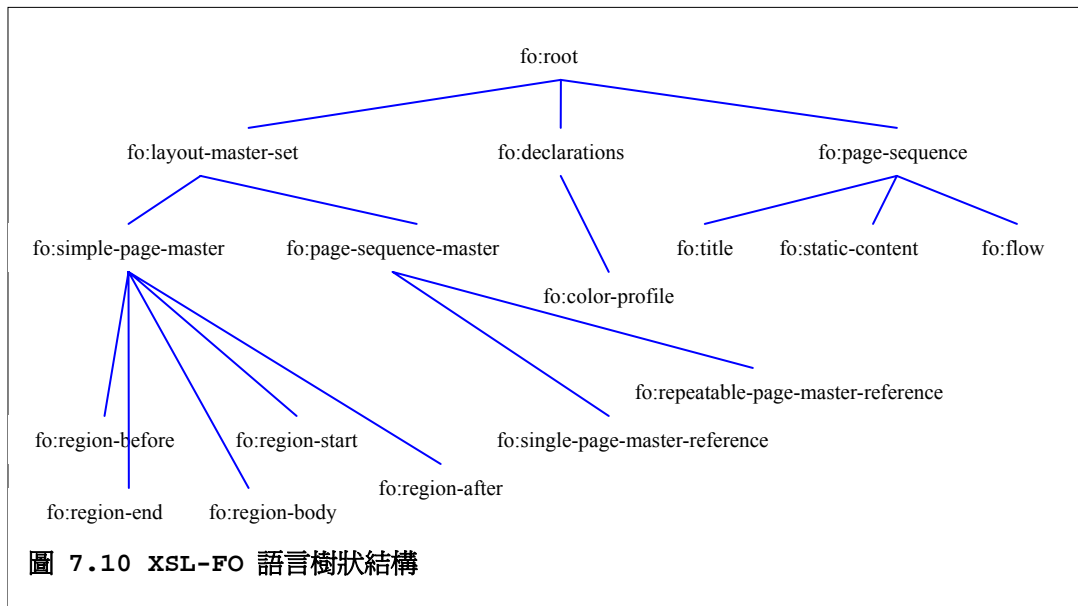
C:\fop-0.20.4>

```

圖 7.9 FOP 命令列執行結果

7.3 XSL-FO 基本觀念

由於 XSL-FO 語言在使用上比起 XSLT 複雜，所以建議每次在學習新的 XSL-FO 元素時，可以參照圖 7.10，該圖是 XSL-FO 元素部份的樹狀結構圖，透過辨認元素所在位置可以加速了解 XSL-FO 語言。



本節會以一份由封面、左頁、右頁所構成的文件作為教導的範例。XSL-FO 檔案就是一份 XML 文件，所以文件的開頭必須 XML Processing Instruction 宣告。接下來就是 XSL-FO 語言的根節點（見圖 7.10） fo:root。所以整份文件一開始應該是如同下面範例所示一般。

```

1 |<?xml version="1.0" encoding="UTF-8"?>
2 |<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3 |</fo:root>

```

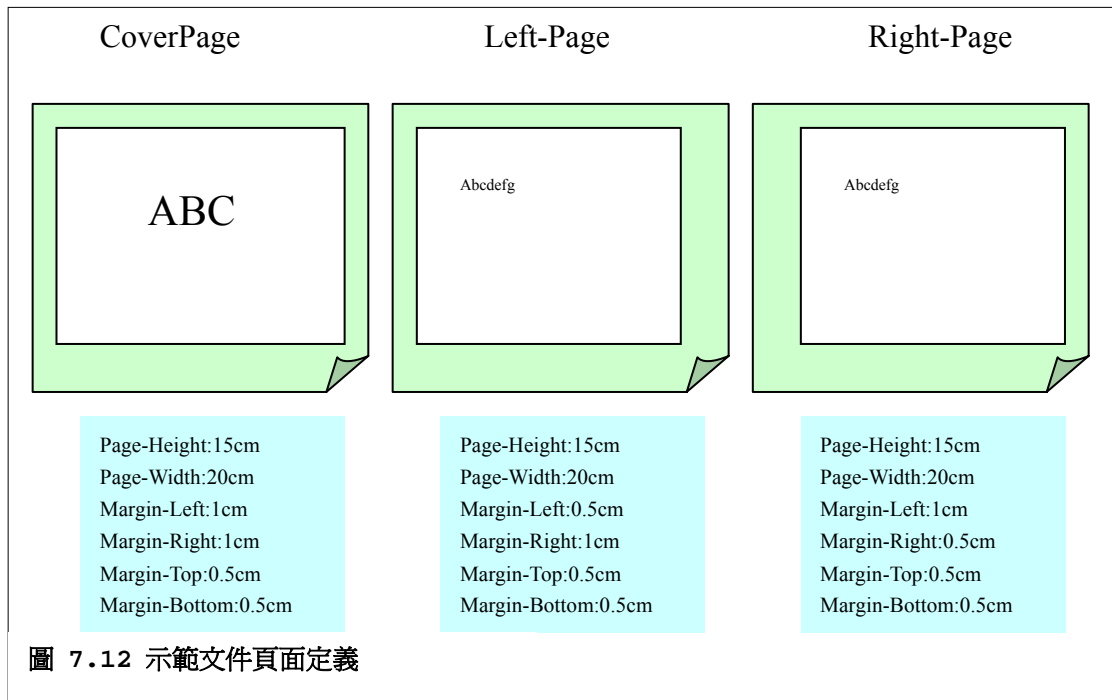
圖 7.11

圖 7.11 的第二行中必須宣告 XSL-FO 的命名空間。

上面的範例是一個空白的 XSL-FO 文件，無法產生任何的目的地文件。而接下來要依序的教導在 <fo:root> 中如何定義文件。

7.3.1 頁面格式 (Page Layout)

在接續 <fo:root> 後緊跟的是頁面格式的定義。在這一節中會以一份以封面、左內頁、右內頁所構成的文件作為範例，說明如何在 XSL-FO 中定義頁面格式。



在圖 7.12 中所見到的文件皆以長度為 20 公分、寬度 15 公分的紙張做為文件輸出時的紙張大小。此外文件頁面的邊界在規劃上也不同，例如首頁 (CoverPage) 在左右邊界個別是留下一公分的邊界，但是左頁 (Left-Page) 和右頁 (Right-Page) 卻因為裝訂邊而在左右邊界設定上不同。

```

1 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
2   <fo:layout-master-set>
3     <fo:simple-page-master master-name="CoverPage"
4       page-height="15cm"
5       page-width="20cm"
6       margin-top="0.5cm"
7       margin-bottom="0.5cm"
8       margin-left="1cm"
9       margin-right="1cm">
10    </fo:simple-page-master>
11    <fo:simple-page-master master-name="LeftPage"
12      page-height="15cm"
13      page-width="20cm"
14      margin-top="0.5cm"
15      margin-bottom="0.5cm"
16      margin-left="0.5cm"
17      margin-right="1cm">
18    </fo:simple-page-master>
19    <fo:simple-page-master master-name="RightPage"
20      page-height="15cm"
21      page-width="20cm"
22      margin-top="0.5cm"
23      margin-bottom="0.5cm"
24      margin-left="1cm"
25      margin-right="0.5cm">
26    </fo:simple-page-master>
27  </fo:layout-master-set>
28 <!-- 其餘省略 -->

```

圖 7.13 頁面格式定義範例

頁面格式的定義的根元素是 `<fo:layout-master-set>`，在圖 7.13 中可以見到 `<fo:layout-master-set>` 中有三個子元素 `<fo:simple-page-master>`。我們以圖 7.13 的第 3 行到第 10 行來說明如何定義一個頁面。

```
<fo:simple-page-master master-name="CoverPage"
  page-height="15cm"
  page-width="20cm"
  margin-top="0.5cm"
  margin-bottom="0.5cm"
  margin-left="1cm"
  margin-right="1cm">
</fo:simple-page-master>
```

`<fo:simple-page-master>` 是用來定義單一頁面格式，透過指定下列屬性達到定義頁面的目的。主要屬性有三大類，第一個是本頁面格式的名稱。在上面的範例中，頁面的名稱為“CoverPage”，也就是封面。第二類屬性就是頁面的大小，透過指定頁面的高度與寬度來定義頁面大小。第三類就是頁面的邊界，邊界一共有上下左右四個屬性，一旦被指定成為邊界，則這些邊界將被視為不可列印的區域。

<code><fo:simple-page-master></code> 屬性說明	
master-name	頁面名稱（之後會被 Page Sequence 參照）
page-height	紙張大小（高度）
page-width	紙張大小（寬度）
margin-top	紙張邊界（上界）
margin-bottom	紙張邊界（下界）
margin-left	紙張邊界（左界）
margin-right	紙張邊界（右界）

圖 7.14

在使用 XSL-FO 時可以使用的測量單位有：

XSL-FO 測量單位	
cm	公分，符合 ISO31 的定義
mm	公釐，符合 ISO31 的定義
in	英吋 2.54cm
pt	1/72 in
pc	12pt
px	像素 (Pixel)
em	相對長度。假設紙張長度是 20cm，而紙張寬度為 1.25em 的意思是 20cm 的 1.25 倍寬，也就是 25 公分寬。

圖 7.15

7.3.2 內容區域 (Content Area)

一旦頁面被定義好時，您必須還要在對頁面可列印區域做更詳盡的定義。XSL-FO 將頁面可列印區域分為五大區域，分別為：Region-Before、Region-After、Region-Start、Region-End、Region-Body，這五大區域的位置可見圖 7.16。

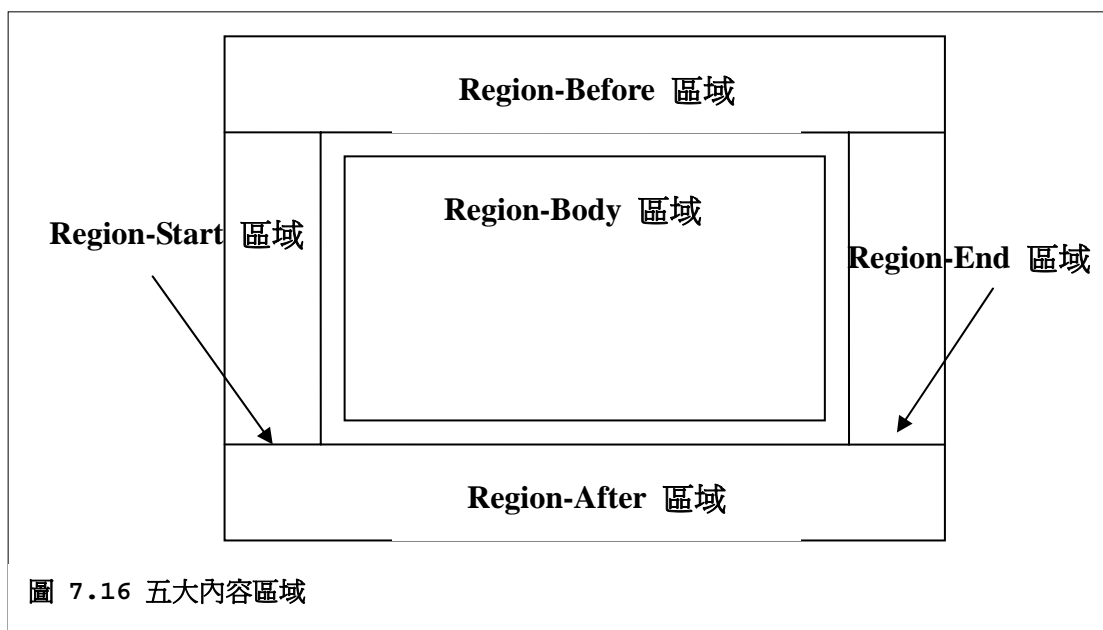


圖 7.16 五大內容區域

每個內容區域中都可以包含不同的本文。例如文件的頁首就是 Region-Before 區域，文件的頁尾就是 Region-After 區域，而 Region-Body 通常是文件本文所在，也是最大的內容區域。

五大內容區域分別由五個不同的元素來定義，這五個元素分別是：

元素名稱	說明
<fo:region-before>	Region Before 區域 (頁首)
<fo:region-after>	Region After 區域 (頁尾)
<fo:region-start>	Region Start 區域 (左頁邊)
<fo:region-end>	Region End 區域 (右頁邊)
<fo:region-body>	Region Body 區域 (本文區)

圖 7.17 五大內容區域元素

定義內容區域的元素必須是 <fo:simple-page-master> 元素之子元素，若在頁面中沒有定義內容區域，則會以預設的方式進行。以封面為例，由於封面不需要頁首頁尾，所以只需要定義 Region-Body。而左右兩內頁必須要定義頁首與頁尾，頁首與頁尾的高度都是一公分。

所以新增內容區域後的 <fo:page-layout-set> 定義變成爲：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <fo:layout-master-set>
```

```

4      <fo:simple-page-master master-name="CoverPage"
5          page-height="15cm"
6          page-width="20cm"
7          margin-top="0.5cm"
8          margin-bottom="0.5cm"
9          margin-left="1cm"
10         margin-right="1cm">
11         <fo:region-body margin-top="4cm" />
12     </fo:simple-page-master>
13     <fo:simple-page-master master-name="LeftPage"
14         page-height="15cm"
15         page-width="20cm"
16         margin-top="0.5cm"
17         margin-bottom="0.5cm"
18         margin-left="0.5cm"
19         margin-right="1cm">
20         <fo:region-before extent="1cm" />
21         <fo:region-after extent="1cm" />
22         <fo:region-body
23             margin-top="1.1cm"
24             margin-bottom="1.1cm" />
25     </fo:simple-page-master>
26     <fo:simple-page-master master-name="RightPage"
27         page-height="15cm"
28         page-width="20cm"
29         margin-top="0.5cm"
30         margin-bottom="0.5cm"
31         margin-left="1cm"
32         margin-right="0.5cm">
33         <fo:region-before extent="1cm" />
34         <fo:region-after extent="1cm" />
35         <fo:region-body
36             margin-top="1.1cm"
37             margin-bottom="1.1cm" />
38     </fo:simple-page-master>
39 </fo:layout-master-set>
40 <!-- 略 -->
41 </fo:root>

```

圖 7.18

所以圖 7.18 中所定義的封面除了紙張尺寸大小外，還定義了本文區，而本文區的限制是本文區上界(margin-top)距離文件頂端 4 公分。而內左頁和內右頁分別定義了頁首（公分）、頁尾（1 公分）、本文區上界與下界(margin-bottom)皆為 1.1 公分。

在此要注意 Region-Body 的上下邊界值必須要大於或是等於 Region-Before 或是 Region-After 的 Extent 屬性質，否則會造成頁首或是頁尾和本文區重疊。

7.3.3 頁面序列 (Page Sequence)

當文件中所需要的版面紙張尺寸都被完整定義後，接下來就要開始定義頁面序列，頁面序列是由多個文件段落所形成的。通常頁面需列必須要對應到特定的頁面格式(Page Sequence)，例如封面就應該對應到封面的頁面格式，內文就應該對應到內文的頁面格式。封面因為只有一頁所以在頁面序列上只需要對應到封

面的頁面格式，沒有特別的問題；而內文因為對應到兩頁（左頁、右頁）所以必須還要進一步的定義，負責定義一個新的頁面格式，這個頁面的格式會自動判別當頁面編號是單數時就使用左頁，而當頁面編號是雙號時，就會使用右頁的頁面格式。

定義單頁文件的頁面格式所使用的元素為 `<fo:simple-page-master>` 而定義多頁文件的頁面格式所使用的元素為 `<fo:page-sequence-master>`，其實 `<fo:pagesequence-master>` 也只是將單頁的頁面格式定義加以歸納而已。例如要定義內頁的格式之前，必須要先定義內頁之左頁和右頁。

```
1 <fo:root>
2   <fo:layout-master-set>
3     <!-- 略 -->
4     <fo:page-sequence-master master-name="Contents">
5       <fo:repeatable-page-master-alternatives>
6         <fo:conditional-page-master-reference
7           master-reference="LeftPage"
8           odd-or-even="even"/>
9         <fo:conditional-page-master-reference
10          master-reference="RightPage"
11          odd-or-even="odd"/>
12       </fo:repeatable-page-master-alternatives>
13     </fo:page-sequence-master>
14   </fo:layout-master-set>
15   <!-- 略 -->
16 </fo:root>
```

圖 7.19 內頁之頁面序列

其中在第 4 行定義這份多頁頁面格式的名稱為 `Contents`，這是為了稍後被參照而給定的。在 `<fo:pagesequence-master>` 元素中的子元素 `<fo:repeatable-pagemaster-alternatives>` 負責迴圈似地挑選符合稍後所指定條件的頁面。

第 6 行與第 9 行分別類似程式語言中的 `Switch` 敘述句，當目前頁面為單數頁面時，就選用左頁當作頁面格式，反之就使用右頁作為頁面格式。

7.3.4 定義頁面內容

定義頁面內容所使用的元素為 `<fo:page-sequence>`，這個元素是由三個子元素所構成，分別是選擇性的 `<fo:title>`，零到多個的 `<fo:static-content>`，和 `<fo:flow>`。

`<fo:title>` 是用來描述本頁的基本資訊，這可以被想像成為文件的摘要部份，或是和 HTML 中的 `<title>` 元素差不多功能，只不過 `<fo:title>` 並不會反映在文件上，這會依據不同的 XSL 處理器與不同的輸出結果有所不同。

`<fo:static-content>` 是用來放置一些不會變動的資訊，例如書名、作者等等。

`<fo:flow>` 是由多個 `<fo:block>` 所構成，而 `<fo:block>` 是段落的意思，也就是說，`<fo:flow>` 是由許多文章段落所形成的。而不同的段落會有不同的字型大小、樣式，這一些段落通通被集合在一起。

7.3.4.1 定義封面內容

封面的內容是由書名（Extensible Stylesheet Language Version 1.0）、圖片、出版日期所構成。大致上呈現的結果如下：

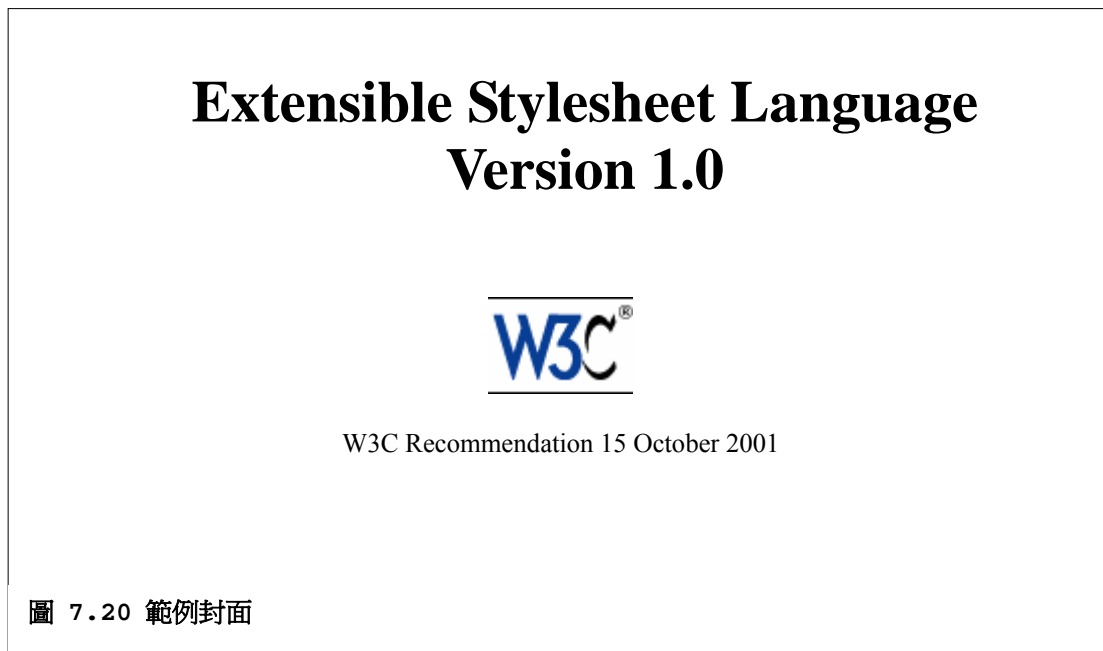


圖 7.20 範例封面

定義封面的片段如下：

```
1 <fo:page-sequence master-reference="CoverPage">
2   <fo:flow flow-name="xsl-region-body">
3     <fo:block font="Arial"
4       font-size="32pt"
5       text-align="center"
6       space-after="14pt">Extensible Stylesheet Language Version 1.0</fo:block>
7     <fo:block text-align="center" space-before="50pt">
8       <fo:external-graphic src="file:w3c_home.bmp"/>
9     </fo:block>
10    <fo:block space-before="1cm"
11      font-family="Times New Roman, Times New Roman"
12      font-size="13pt"
13      text-align="center">W3C Recommendation 15 October 2001
14    </fo:block>
15  </fo:flow>
16 </fo:page-sequence>
```

圖 7.21 封面內容定義

在圖 7.21 的第 1 行對於頁面內容的定義參照了頁面格式“CoverPage”，所以頁面內容會依據“CoverPage”的定義做安排。在第 2 行定義了一個 `<fo:flow>`，由於頁面中有五大內容區域（Content Area），所以必須透過屬性 `flow-name` 來指定 `<fo:flow>` 是屬於哪一個內容區域；所以一共擁有五個 `flow-name`，分別為 `xsl-region-body`, `xsl-region-before`, `xsl-region-after`, `xsl-region-start`, `xsl-region-end`。

在圖 7.21 中的 `<fo:flow>` 下擁有三個 `<fo:block>`，第一個 `<fo:block>` (第 3 行) 敘述了這個段落所使用的字體是 “Arial”，字體大小 (font-size) 為 32pt，文字置中對齊，本段落後留白 14pt，段落的內文為「Extensible Stylesheet Language Version 1.0」。

第二個 `<fo:block>` 是一張圖片，這張圖片也是透過指定 `text-align="center"` 來達到圖片置中對齊的需求。而 `<fo:block>` 中的子元素 `<fo:external-graphic>` 指出了外部圖片檔案的來源。

第三個 `<fo:block>` 就不再多說了。由於 `<fo:block>` 的屬性非常的多，礙於篇幅無法一一闡述，詳細請查閱 XSL 規格書 (<http://www.w3.org/TR/xsl/>)。

以下是使用 FOP 所產生出的 PDF 檔案結果：

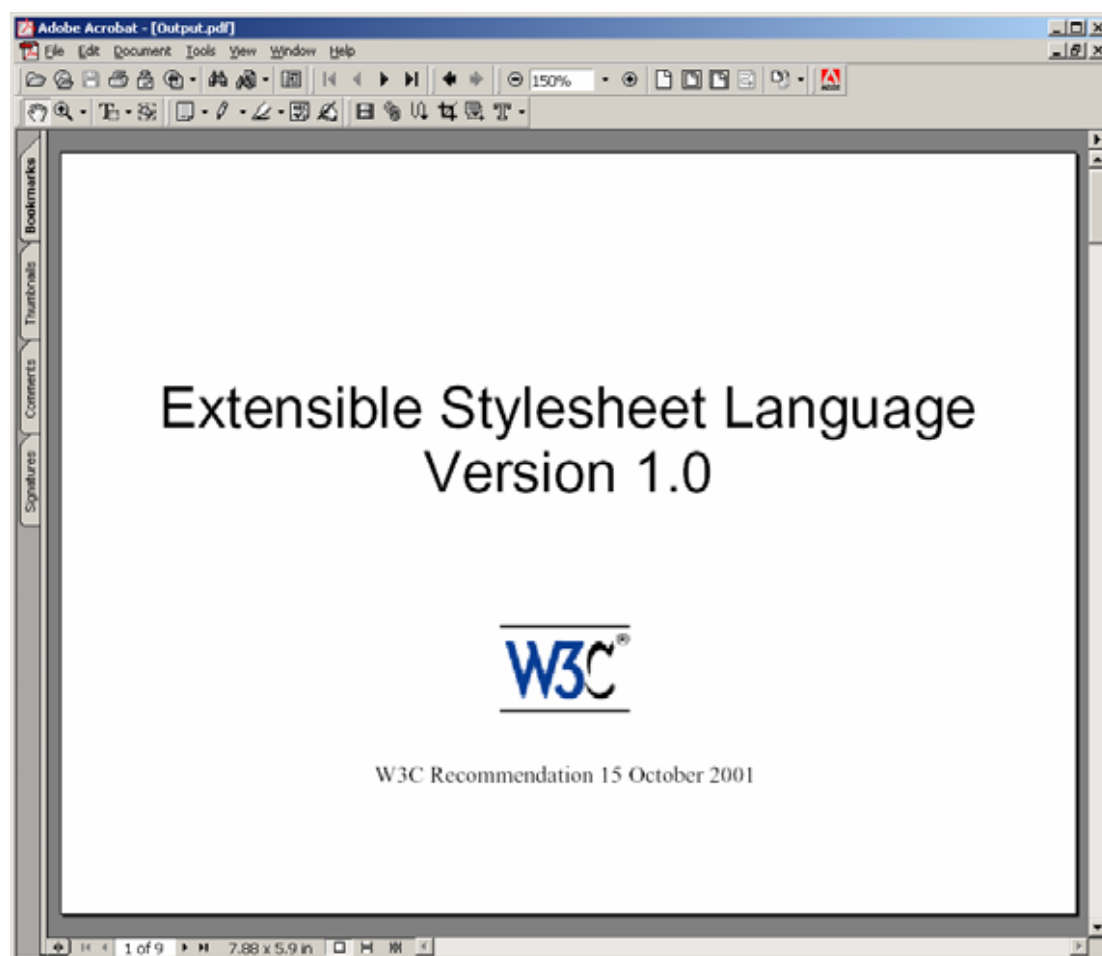


圖 7.22 封面的輸出結果

7.3.4.2 定義內頁內容

構成內頁的段落與內容比封面來得多很多，不過方法還是一樣。首先定義 `<fo:page-sequence>` 元素並透過屬性 `master-reference` 參照到 `<fo:layout-master-set>` 中所定義的內頁格式 `<fo:page-sequence-master`

master-name="CoverPage">。然後建立 <fo:title> 建立頁面的基本摘要，使用 <fo:static-content> 用來處理頁首頁尾的資訊，最後建立 <fo:flow> 與其段落子元素 <fo:block>

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <!-- 略 -->
4   <fo:page-sequence master-reference="Contents"
initial-page-number="2">
5     <fo:static-content flow-name="xsl-region-before">
6       <fo:block font-size="10pt"
7         text-align="center">Extensible Stylesheet Language Version 1.0
8       </fo:block>
9     </fo:static-content>
10    <fo:static-content flow-name="xsl-region-after">
11      <fo:block font-size="10pt" text-align="center">Page
#<fo:page-number/>
12      </fo:block>
13    </fo:static-content>
14    <fo:flow flow-name="xsl-region-body">XSL 1.0 (XSL:FO)
15      <fo:block font-size="16pt" font="Times New Roman"
16        font-weight="bold" color="blue"
17        space-before="10pt" space-after="10pt">
18        Extensible Stylesheet Language 1.0
19      </fo:block>
20      <fo:block font-size="14pt" font="Times New Roman"
21        font-weight="bold" space-before="14pt"
22        space-after="10pt">1.Introduction and Overview
23      </fo:block>
24      <fo:block text-indent="1.2cm" font="Times New Roman"
font-size="10">
25        This specification defines the Extensible Stylesheet Language
26        (XSL). XSL is a language for expressing stylesheets. Given a
27        class of arbitrarily structured XML [XML] documents or data files,
28        designers use an XSL stylesheet to express their intentions about
29        how that structured content should be presented; that is, how the
30        source content should be styled, laid out, and paginated onto some
31        presentation medium, such as a window in a Web browser or a
32        hand-held device, or a set of physical pages in a catalog, report,
33        pamphlet, or book.
34      </fo:block>
35    </fo:flow>
36  </fo:page-sequence>
37 </fo:root>

```

圖 7.23 內頁部份內容定義

在 `<fo:page-sequence>` 元素中的屬性 `initial-page-number="2"` 定義了內頁起始的號碼，這個號碼會反應在頁尾的“Page #頁數”資訊上。

在內頁中和封面不同的是定義了頁首與頁尾。請看頁首的定義：

```

<fo:static-content flow-name="xsl-region-before">
  <fo:block font-size="10pt"
    text-align="center">Extensible Stylesheet Language
Version 1.0
  </fo:block>
</fo:static-content>

```

頁首的部份在定義 `<fo:static-content>` 元素時設定了所定義的內容區域為“xsl-region-before”，這就是頁首的區域。在這個區域中，有一個段落 `<fo:block>` 定義了字體大小為 10pt 且置中對齊的文字“Extensible Stylesheet Language Version 1.0”。所以往後所有的內頁，不論是左頁還是右頁的頁首都會出現該段落文字。

而頁尾和頁首不一樣的地方在於宣告 `<fo:static-content>` 所設定的 `flow-name` 不同。

```
<fo:static-content flow-name="xsl-region-after">
  <fo:block font-size="10pt" text-align="center">
    Page #<fo:page-number/>
  </fo:block>
</fo:static-content>
```

此外在 `<fo:block>` 中使用了 `<fo:page-number>`，這個元素會在產生文件時計算目前該頁的頁碼為何。

接下來就是定義 `<fo:flow>`，而`<fo:flow>` 中變化最多的就是 `<fo:block>`，藉由指定 `<fo:block>` 元素中特定的屬性以調整字體大小、文字擺放位置、縮排、顏色還有前後段落來達到排版的效果。對 `<fo:block>` 的介紹請參閱 7.4.1。

以下是內頁產生出來的結果：

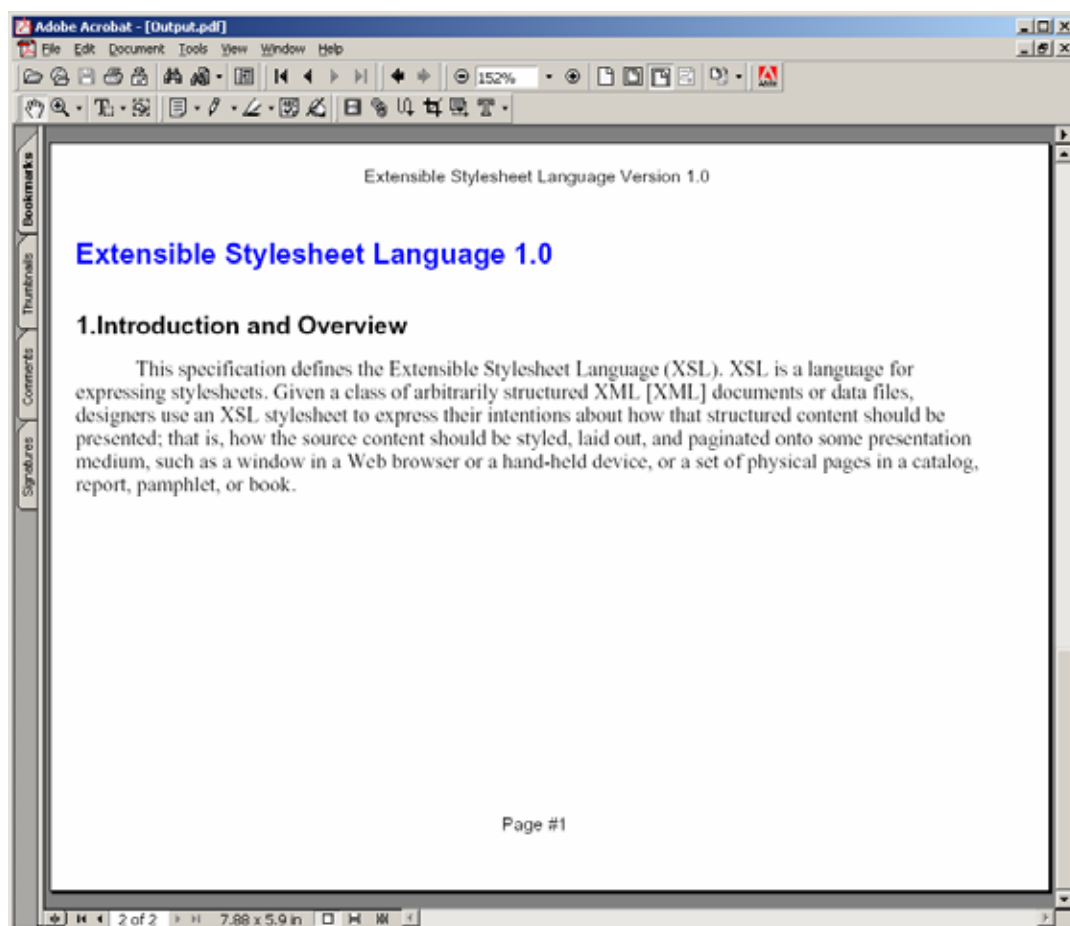


圖 7.24 內頁的輸出結果

7.4 XSL-FO 進階技術

在這一節中將要討論進階的應用，例如表格、清單、內文樣式等等。在熟悉 XSL-FO 語彙之後方可使用 XSLT 將 XML 或是 HTML 轉換成複雜的 XSL-FO 語言了。

7.4.1 建立段落使用 <fo:block> 元素

<fo:block> 元素是用於格式化段落 (Paragraphs)、文件抬頭(Titles)、標題 (Headlines)、圖表(Figure)、表格等等，而 <fo:block> 中的子元素可以是 #PCDATA、<fo:inline>、或是<fo:block>。

常用的 <fo:block> 屬性可以分爲下列幾大類，以表格歸納如下：

<fo:block> 常用字體屬性	
font-family	字型家族，例如 "sans-serif"
font-weight	數值越大，字體越厚重。例如 "100" 的加權會讓字體看起來比較飄逸；而 "900" 就會讓字體看起來厚重。
font-style	字體樣式，例如 斜體
font-size	字體大小

<fo:block> 常用背景屬性	
background-color	背景顏色
background-image	背景影像
background-repeat	重複背景影像（當背景影像不夠填滿段落時）
background-attachment	設定影像與背景結合方式

<fo:block> 常用邊框屬性	
border-location-info	
其中 <i>location</i> 有 <i>before</i> , <i>after</i> , <i>start</i> , <i>end</i> , <i>top</i> , <i>bottom</i> , <i>left</i> , 或是 <i>right</i>	
而 <i>info</i> 有 <i>style</i> , <i>width</i> , <i>color</i> 三種資訊。	

所以 border-start-color 屬性就是當書寫方向是由左而右時，左邊邊框的顏色。

<fo:block> 常用邊緣填充屬性	
border-location-info	
其中 <i>location</i> 有 <i>before</i> , <i>after</i> , <i>start</i> , <i>end</i> , <i>top</i> , <i>bottom</i> , <i>left</i> , 或是 <i>right</i>	
所以 padding-start 屬性就是當書寫方向是由左而右時，左邊填充留白寬度。	

<fo:block> 常用邊界屬性

margin-location

其中 *location* 有 *top*, *bottom*, *left*, 或是 *right*

所以 *margin-top* 就是指上邊界有多寬 (長)。

<fo:block> 常用文字定位屬性

text-align	文字對齊方式，例如文字置中對齊。
text-align-last	和上面同，只是給 <fo:block> 中最後一行文字使用。
屬性值	<i>start</i> , <i>end</i> , <i>left</i> , <i>right</i> , <i>center</i>

<fo:block> 常用文字縮排屬性

text-indent	第一行文字縮排寬度，例如 1cm。
start-indent	距離內文上邊界縮排寬度
end-indent	距離內文下邊界縮排寬度

<fo:block> 常用與前後物件間距屬性

space-before.minimum	與上個格式化物件最小距離
space-before.optimum	與上個格式化物件最適當距離
space-before.maximum	與上個格式化物件最大距離
space-before	與上個格式物件距離
space-after.minimum	與下個格式化物件最小距離
space-after.optimum	與下個格式化物件最適當距離
space-after.maximum	與下個格式化物件最大距離
space-after	與下個格式物件距離

下圖 7.25 是一段 <fo:block> 範例：

```
1 <fo:block
2   text-indent="1cm"
3   font-family="sans-serif"
4   font-size="12pt"
5   space-before.minimum="2pt"
6   space-before.maximum="6pt"
7   space-before.optimum="4pt"
8   space-after.minimum="2pt"
9   space-after.maximum="6pt"
10  space-after.optimum="4pt"
11  text-align="center" >
12 The XSL comes in two parts,
13 one is XSL-Transformation ,
14 and another is XSL-Formatting Objects
15 </fo:block>
```

圖 7.25

圖.25 的 <fo:block> 屬性定義了字串「**The XSL comes in two parts, one is XSL-Transformation , and another is XSL-Formatting**

Objects」在文件段落中首行縮排一公分（圖 7.25 第 2 行），使用的字體是 sans-serif（圖 7.25 第 3 行），字體的大小為 12pt（圖 7.25 第 4 行），與前面物件（可能是上一個 `<fo:block>` 或是表格、清單等等物件）間隔最小 2pt、最大 6pt、最適 4pt（圖 7.25 第 5~7 行），雨後面物件間隔最小 2pt、最大 6pt、最適 4pt（圖 7.25 第 8~10 行），文字置中對齊（圖 7.25 第 11 行）。

7.4.2 內文樣式 `<fo:inline>`

內文樣式適用於強調文章中特定關鍵字或是關鍵句子。透過將文字臨時以不同字型、字體來達成強調的效果。

```
1 <fo:block
2   text-indent="1.2cm" font-size="10">
3   This specification defines the Extensible Stylesheet Language
4   (XSL). XSL is a language for expressing stylesheets. Given a class of
5   arbitrarily structured
6   <fo:inline font-style="italic"
7     font-weight="800"
8     color="red"
9     text-decoration="underline">XML documents
10  </fo:inline>
11  or data files, designers use
12  an XSL stylesheet to express their intentions about how that
13  structured content should be presented; that is, how the source
14  content should be styled, laid out, and paginated onto some
15  presentation medium, such as a window in a Web browser or a
16  hand-held device, or a set of
17  physical pages in a catalog, report, pamphlet, or book.
18 </fo:block>
```

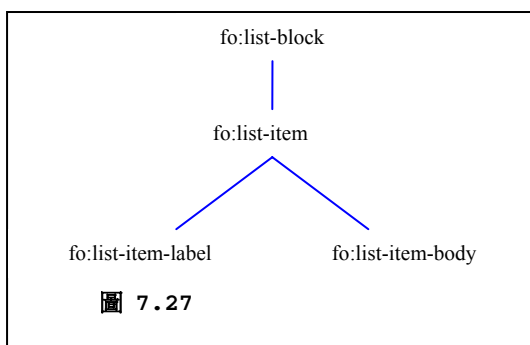
圖 7.26 `<fo:inline>` 範例

在圖 7.26 中的第 10 行宣告了一個 `<fo:inline>` 元素，這個元素是要強調關鍵字「XML Documents」，讓這個關鍵字的字體以斜 (`font-style="italic"`)、粗體 (`font-weight="800"`)、底線 (`font-decoration="underline"`)、紅色 (`color="red"`) 來表示。

`<fo:inline>` 和 `<fo:block>` 有許多共同使用的屬性集合，例如字體類的屬性、前景背景樣式屬性等等。`<fo:inline>` 可以功能和 HTML 中的 ``，或是 `<i>` 元素相似，只不過 `<fo:inline>` 強調的不是單一樣式，而是透過設定不同的屬性已符合文件段落中所需要的樣式。

7.4.3 建立清單(List)

在 XSL-FO 中建立清單必須使用下列元素：`<fo:list-block>`、`<fo:list-item>`、`<fo:list-item-label>`、`<fo:list-item-body>`。這些元素的階層關係如下圖：



<fo:list-block> 是建立清單系列元素的根元素，這個元素的組成是由一到多個的 <fo:list-item> 所構成，<fo:list-item> 元素就是清單中的一則項目，而項目是由項目標籤<fo:list-item-label> 與項目內容 <fo:list-item-body> 所構成。在這兩個元素都必須有一個 <fo:block> 元素。

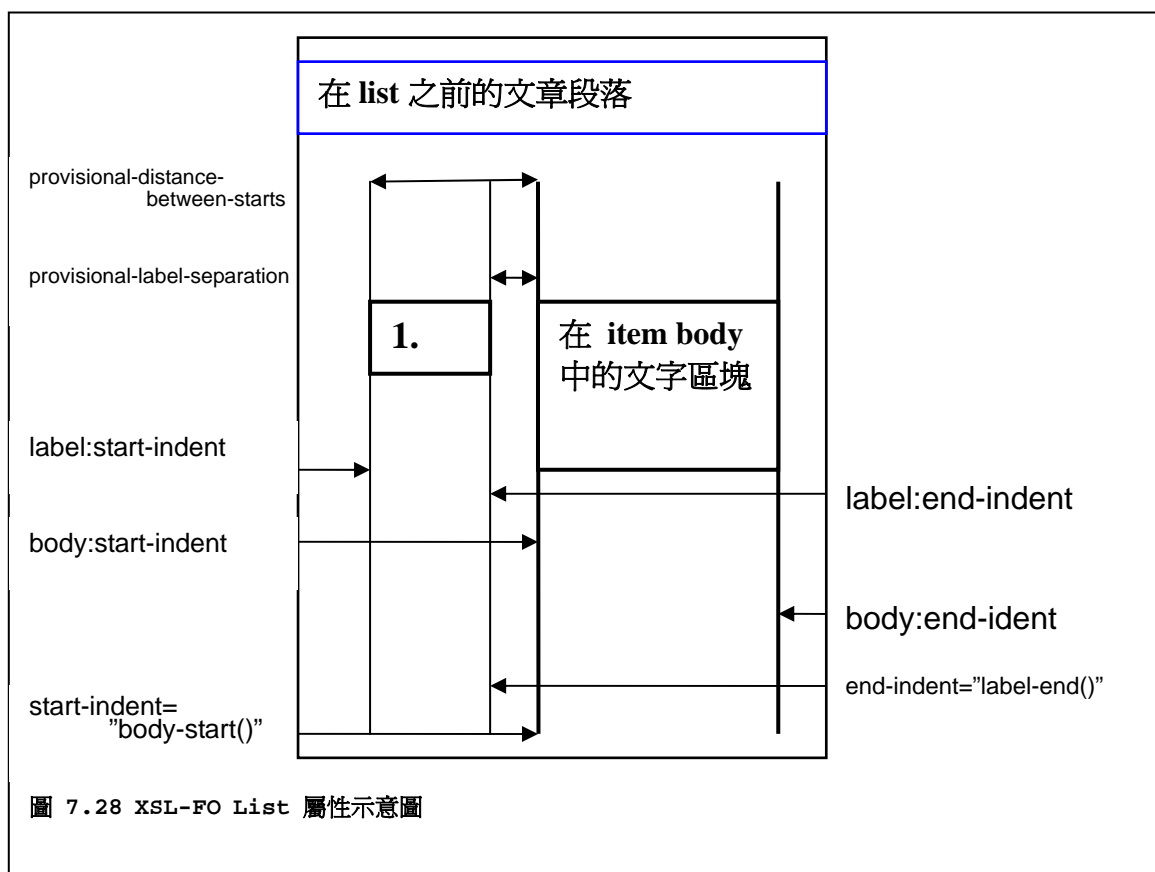


圖 7.28 是建立清單所需要瞭解的幾個重要屬性。
 provisional-distance-between-starts 屬性與 provisional-label-separation 屬性皆是定義於 <fo:list-block> 之中。在圖 7.28 中可以見到
 provisional-distance-between-starts 表示清單標籤到 Item Body 之間的距離；而
 provisional-label-separation 是清單標籤與清單內容之間的距離。

屬於 <fo:label> 中的屬性有 start-indent 與 end-indent 兩個屬性。
 start-indent 屬性表示元素在段落文字邊界開始到清單內容前的距離，使用者可以直接指定距離，也可以使用 body-start() 函數，這個函數會參照到 <fo:list-block> 中的 provisional-distance-between-starts 屬性與 provisional-label-separation 屬

性，並傳回適當的值。而 `end-indent` 則剛好和 `start-indent` 相反，其距離是由清單標籤到段落文字邊界尾端為止。使用者也可以使用 `label-end()` 函數來簡化計算清單內容尾部縮排的寬度。

下面將建立一個清單，這個清單的項目分別是：

The following items are PC hardware components.

- Item #1. Computer Case.
- Item #2. Power Supplier.
- Item #3. Mother Board.
- Item #4. Fans.
- Item #5. Hard Disks.

完整的 XSL-FO 範例如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <fo:layout-master-set>
4     <fo:simple-page-master master-name="PageA4"
5       page-height="29.7cm"
6       page-width="21cm"
7       margin-top="2.5cm"
8       margin-bottom="3.5cm"
9       margin-left="2.5cm"
10      margin-right="2.5cm">
11       <fo:region-body/>
12     </fo:simple-page-master>
13   </fo:layout-master-set>
14   <fo:page-sequence master-reference="PageA4">
15     <fo:flow flow-name="xsl-region-body">
16       <fo:block font-size="16pt" font-weight="bolder" >
17         The following items are PC hardware components.
18       </fo:block>
19       <fo:list-block space-before="1cm"
20         provisional-distance-between-starts="3cm"
21         provisional-label-separation="0.4cm"
22         start-indent="1cm" >
23         <fo:list-item >
24           <fo:list-item-label >
25             <fo:block>Item #1.</fo:block>
26           </fo:list-item-label>
27           <fo:list-item-body start-indent="body-start()">
28             <fo:block>Computer Case.</fo:block>
29           </fo:list-item-body>
30         </fo:list-item>
31         <fo:list-item >
32           <fo:list-item-label >
33             <fo:block>Item #2.</fo:block>
34           </fo:list-item-label>
35           <fo:list-item-body start-indent="body-start()">
36             <fo:block>Power Supplier.</fo:block>
37           </fo:list-item-body>
38         </fo:list-item>
```

```

39         <fo:list-item >
40             <fo:list-item-label >
41                 <fo:block>Item #3.</fo:block>
42             </fo:list-item-label>
43             <fo:list-item-body start-indent="body-start()" >
44                 <fo:block>Mother Board.</fo:block>
45             </fo:list-item-body>
46         </fo:list-item>
47         <fo:list-item >
48             <fo:list-item-label >
49                 <fo:block>Item #4.</fo:block>
50             </fo:list-item-label>
51             <fo:list-item-body start-indent="body-start()" >
52                 <fo:block>Fans.</fo:block>
53             </fo:list-item-body>
54         </fo:list-item>
55         <fo:list-item >
56             <fo:list-item-label >
57                 <fo:block>Item #5.</fo:block>
58             </fo:list-item-label>
59             <fo:list-item-body start-indent="body-start()" >
60                 <fo:block>Hard Disks.</fo:block>
61             </fo:list-item-body>
62         </fo:list-item>
63     </fo:list-block>
64 </fo:flow>
65 </fo:page-sequence>
66 </fo:root>

```

圖 7.29 DemoList.fo

在 DemoList.fo 中宣告了一個 A4 大小的頁面，第一行的文字字體為 16pt。清單與第一行之間間距為 1cm，整份清單縮排 1cm。在清單的定義中 provisional-distance-between-starts 屬性為 3cm，而 provisional-label-separation 的屬性為 0.4cm。在 <fo:list-item-label> 與 <fo:listitem-body> 中的 startindent 與 end-indent 分別交由函數 body-start() 與 label-end() 來完成。

下面是 DemoList.fo 產生的 PDF 檔案畫面。

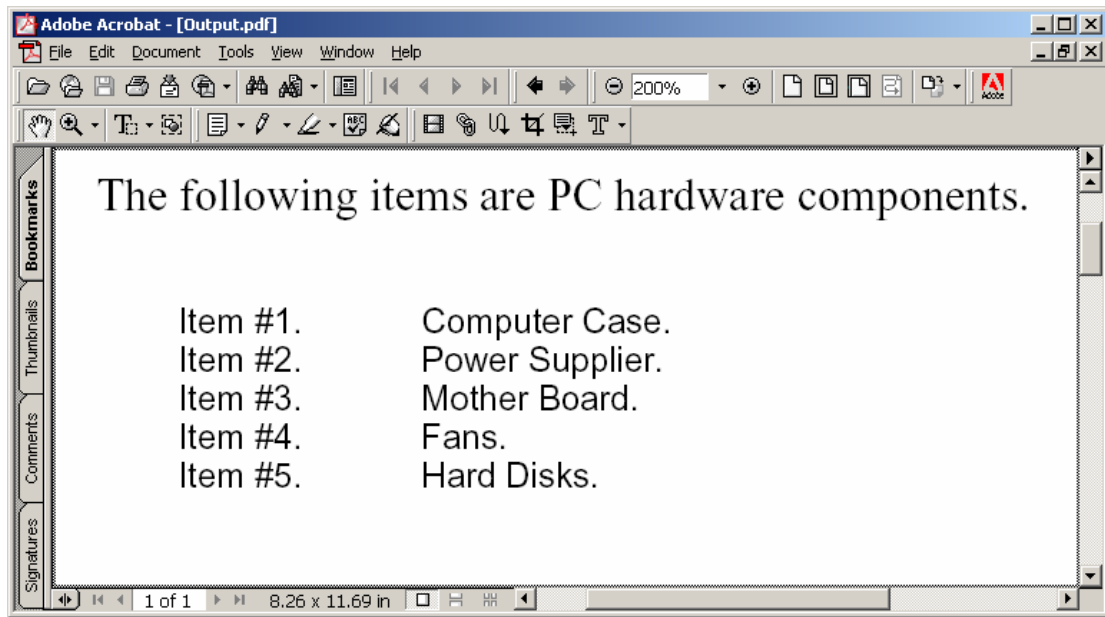


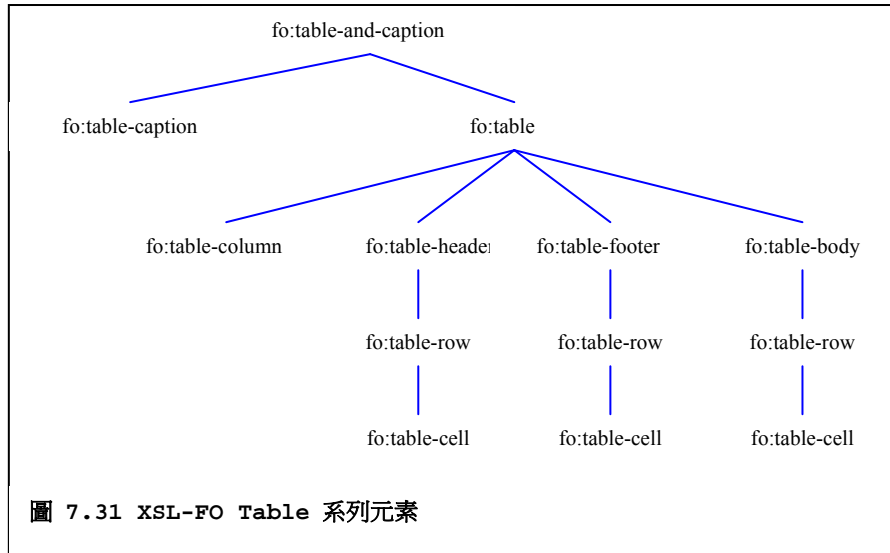
圖 7.30 內頁的輸出結果

7.4.4 建立表格

在 XSL-FO 建立表格所需要的元素有：

```
<fo:table-and-caption>  
<fo:table-caption>  
<fo:table>  
<fo:table-column>  
<fo:table-header>  
<fo:table-footer>  
<fo:table-body>  
<fo:table-row>  
<fo:table-cell>
```

這些元素的階層關係如下圖：



`<fo:table-and-caption>` 是定義表格的根元素，在 `<fo:table-and-caption>` 下有兩個子元素分別為 `<fo:table-caption>` 和 `<fo:table>`。 `<fo:table-caption>` 元素是用來定義表格的標題，而 `<fo:table>` 就是詳細的表格定義。

下面表格是 `<fo:table>` 所有子元素的功能說明：

<fo:table> 子元素說明	
<fo:table-column>	定義表格的欄位，例如欄位寬度、欄位號碼
<fo:table-header>	表格的表頭
<fo:table-footer>	表格註腳
<fo:table-row>	表格的列，在此元素中包含多個 <code><fo:table-cell></code>
<fo:table-cell>	表格內的儲存格

下圖 7.32 的範例定義了兩欄三列的表格。

```

1 <fo:table-and-caption>
2   <fo:table-caption>
3     <fo:block>The Caption Of Table</fo:block>
4   </fo:table-caption>
5   <fo:table>
6     <fo:table-column column-width="150pt"/>
7     <fo:table-column column-width="150pt"/>
8     <fo:table-body font-size="10pt" font-family="sans-serif">
9       <fo:table-row>
10        <fo:table-cell>
11          <fo:block>text</fo:block>
12        </fo:table-cell>
13        <fo:table-cell>
14          <fo:block>text</fo:block>
15        </fo:table-cell>
16      </fo:table-row>
17      <fo:table-row>
18        <fo:table-cell>
19          <fo:block>text</fo:block>
20        </fo:table-cell>
21        <fo:table-cell>
22          <fo:block>text</fo:block>
  
```

```

23         </fo:table-cell>
24     </fo:table-row>
25     <fo:table-row>
26         <fo:table-cell>
27             <fo:block>text</fo:block>
28         </fo:table-cell>
29         <fo:table-cell>
30             <fo:block>text</fo:block>
31         </fo:table-cell>
32     </fo:table-row>
33 </fo:table-body>
34 </fo:table>
35 <fo:table-and-caption>

```

圖 7.32 XSL-FO 表格使用範例

7.5 使用 XSLT 產生 XSL-FO

7.5.1 由 XML 產生 XSL-FO

XML 文件本身的結構並不適合列印，透過 XSLT(Extensible Stylesheet Language Transformation) 可以將 XML 文件轉換成 HTML 或是其他純文字型態格式檔案。在本節中將利用 XSLT 將 XML 文件轉換成爲 XSL-FO 語言，再交由 XSL-FO 處理器輸出到 PDF 檔案。

在 7.5.1 中使用的是在前面章節所使用過的學生成績列表的範例，範例內容如下：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <?xml-stylesheet type="text/xsl" href="ScoreCard2Fo.xslt"?>
3 <ScoreOfStudent xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xsi:noNamespaceSchemaLocation="ScoreCard.xsd">
5     <ScoreCardInScienceClass
6         Name="Mahler" Date="2002-08-21"
7         StudentId="41003" HaveLab="true"
8         ClassId="Class A" MailHome="true">
9         <English>97</English>
10        <Math>68</Math>
11        <Science>86</Science>
12        <Physics>75</Physics>
13        <Biology>92</Biology>
14    </ScoreCardInScienceClass>
15    <ScoreCardInClassicClass Name="Jason"
16        Date="2002-08-21" StudentId="42001"
17        ClassId="Class C" MarketResearch="true"
18        MailHome="true">
19        <English>56</English>
20        <Math>61</Math>
21        <Sociology>82</Sociology>
22        <History>92</History>
23        <Music>97</Music>
24    </ScoreCardInClassicClass>
25    <ScoreCardInScienceClass Name="Peter"

```

```

26         Date="2002-08-20" StudentId="41004"
27         HaveLab="true" ClassId="Class A"
28         MailHome="true">
29             <English>67</English>
30             <Math>34</Math>
31             <Science>66</Science>
32             <Physics>33</Physics>
33             <Biology>54</Biology>
34     </ScoreCardInScienceClass>
35     <ScoreCardInClassicClass Name="Jessica"
36         Date="2002-04-13" StudentId="42004"
37         ClassId="Class C" MarketResearch="true"
38         MailHome="true">
39         <English>100</English>
40         <Math>54</Math>
41         <Sociology>44</Sociology>
42         <History>59</History>
43         <Music>100</Music>
44     </ScoreCardInClassicClass>
45 </ScoreOfStudent>

```

圖 7.33 ScoreCard.xml

在建立 ScoreCard.xml 產生 XSL-FO 語言的樣式表 (Stylesheet) 之前有下列幾個步驟要進行：

1. 分析 XML 文件中要安排輸出到 XSL-FO 的各項元素與屬性。
2. 模擬輸出的版面配置。(例如紙張大小、邊界、頁首頁尾等)。
3. 依據模擬輸出的版面配置，設計 XSL-FO 範本。
4. 依據已經設計好的 XSL-FO 範本設計 XSLT 所需要的樣式表。
5. 使用 XSLT 產生目的 XSL-FO，並使用 XSL-FO 處理器產生輸出結果。

第一個步驟、分析 XML 文件

ScoreCard.xml 是用來記錄學生的成績的 XML 檔案。在這一份 XML 檔案中，學生被分成兩種班別，在不同的班別中所修習的科目不同。用來記錄這兩個班別學生個人成績的元素分別是 `<ScoreCardInScienceClass>` 與 `<ScoreCardInClassicClass>`，前者是理科專班，後者是文科專班。在這兩類元素中皆有共同的屬性用來紀錄學生的基本資料，這些屬性分別是姓名 (Name)、考試日期 (Date)、學號 (StudentID)、班別 (ClassID)、是否郵寄成績單回家 (MailHome)。另外理科專班還多了一個屬性 HaveLab，用來記錄學生是否有實驗室；而文科專班則有 MarketResearch 屬性用來紀錄是否有進行市場調查。本次輸出只需要兩種不同班別學生的共同資料，其他非共通的學生基本資料，留置不做輸出使用。

在 `<ScoreCardInScienceClass>` 與 `<ScoreCardInClassicClass>` 中的子元素皆是各科目的成績，成績皆是由 `<科目名稱>分數</科目名稱>` 所構成，例如英文成績的標記範例為 `<English>94</English>`。所以 ScoreCard.xml 將會選擇學生基本資料與各科成績作為輸出的資料來源。

第二個步驟、模擬版面配置

在設計 XSL-FO 之前，若先行將文件輸出的版面進行配置模擬配置，可以省去許多時間調整 XSL-FO 在版面上的設定，另外版面模擬的越詳細，在設計 XSL-FO 所花的時間也能大幅的減少。下圖是 ScoreCard.xml 輸出的模擬版面配置。

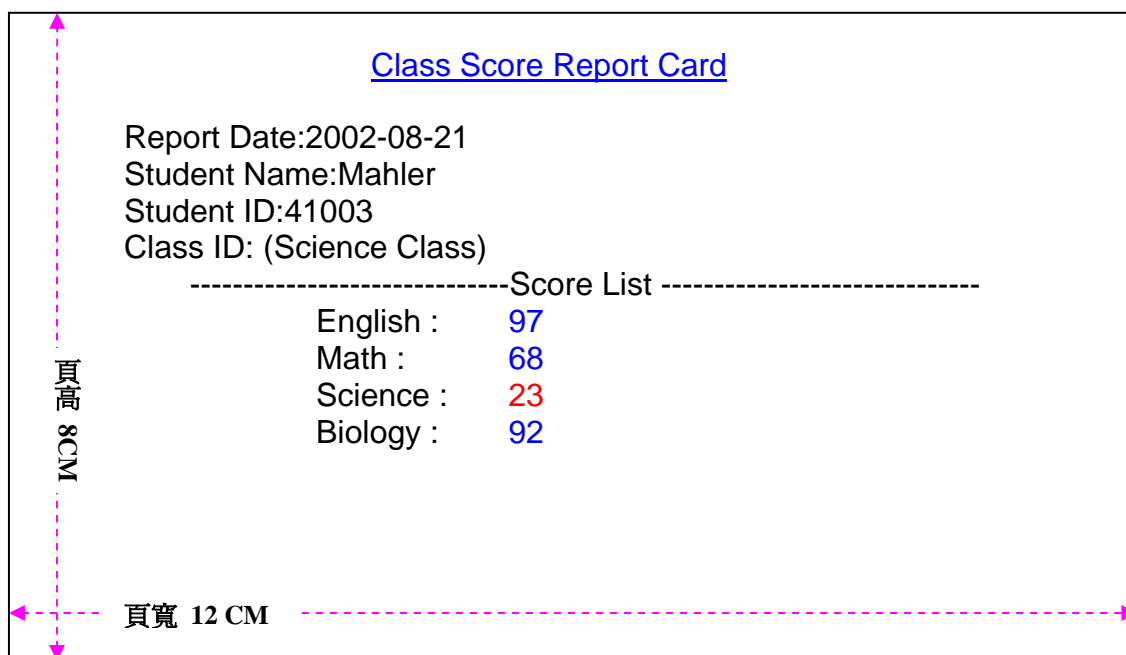


圖 7.34 模擬版面配置

圖 7.34 中的版面配置是模擬學生成績卡的輸出規格，這種成績卡的紙張的長度與寬度分別為 8cm 與 12cm，在頁面的邊界上，四個角落分別都是 0.5cm 的邊界。頁面中的定義了兩塊內容區域，分別為頁首 (xsl-region-before) 與內容 (xsl-region-body)。頁首的高度為 1cm，內容的上邊界為 0.5cm 而下邊界為 0.8cm。

在版面配置上可以清楚的見到所要輸出的資料項目。在輸出頁面內容的上半部是學生的基本資料，如學號、日期、姓名等等，頁面中間是一條分隔線，在頁面下方則是各科的成績，及格的分數顏色為藍色，而不及格的分數顏色為紅色。

第三個步驟、設計 XSL-FO 範本

當版面配置模擬完成後，可以直接將版面配置直接套用在設計 XSL-FO 之上。例如頁面的基本資訊馬上就可以轉換成下面的 XSL-FO 片段。

```
<fo:layout-master-set>
  <fo:simple-page-master master-name="Card"
    page-height="8cm"
    page-width="12cm"
    margin="0.5cm 0.5cm 0.5cm 0.5cm">
    <fo:region-body margin-top="0.5cm" margin-bottom="0.8cm"/>
    <fo:region-before extent="1cm"/>
  </fo:simple-page-master>
</fo:layout-master-set>
```

當頁面基本資訊設計完成後，接下來的就是設計文件的內容。第一個部份是頁首，頁首的資訊由於都是固定不變的，所以使用 <fo:static-content> 元素來顯示頁首文字。

```
<fo:static-content flow-name="xsl-region-before">
  <fo:block text-align="center" font-size="10pt" color="blue">
    <fo:inline text-decoration="underline">
      Class Score Report Card
    </fo:inline>
  </fo:block>
</fo:static-content>
```

依據先前版面配置的資訊，頁首的文字必須是置中對齊，而且是藍色有底線的文字「Class Score Report Card」。

頁面內容的資訊是由學生基本資料、分隔線、成績所構成，第一個部份是學生的基本資料，這些資料只需要由幾個單純的 <fo:block> 所組成即可。

```
<fo:block font-size="12pt">Report Date:2002-08-21</fo:block>
<fo:block font-size="12pt">Student Name:Mahler</fo:block>
<fo:block font-size="12pt">Student ID:41003</fo:block>
<fo:block font-size="12pt">Class ID: (Science Class)</fo:block>
```

此外分隔學生基本資料與學生成績的分隔線也是很容易完成，只需要一個 <fo:block> 元素，但是該元素必須設定前後段落的距離。

```
<fo:block space-before="0.3cm" text-align="center">
-----Score List
-----
</fo:block>
```

最後一個比較複雜的是學生的成績，學生的成績是由 <fo:list-block> 所構成，由科目的名稱做為清單的標籤，科目成績作為清單的項目。另外若成績低於六十分者，以紅色顯示，否則以藍色顯示。

```
<fo:list-block space-before="0.2cm"
provisional-distance-between-starts="3cm"
provisional-label-separation="0.4cm"
start-indent="1cm">
<fo:list-item>
  <fo:list-item-label>
    <fo:block>English :</fo:block>
  </fo:list-item-label>
  <fo:list-item-body start-indent="body-start()">
    <fo:block color="blue">97</fo:block>
  </fo:list-item-body>
</fo:list-item>
<fo:list-item>
  <fo:list-item-label>
    <fo:block>Math :</fo:block>
  </fo:list-item-label>
  <fo:list-item-body start-indent="body-start()">
    <fo:block color="blue">68</fo:block>
  </fo:list-item-body>
</fo:list-item>
<fo:list-item>
  <fo:list-item-label>
```

```

        <fo:block>Science :</fo:block>
      </fo:list-item-label>
      <fo:list-item-body start-indent="body-start()">
        <fo:block color="blue">86</fo:block>
      </fo:list-item-body>
    </fo:list-item>
    <fo:list-item>
      <fo:list-item-label>
        <fo:block>Physics :</fo:block>
      </fo:list-item-label>
      <fo:list-item-body start-indent="body-start()">
        <fo:block color="blue">75</fo:block>
      </fo:list-item-body>
    </fo:list-item>
    <fo:list-item>
      <fo:list-item-label>
        <fo:block>Biology :</fo:block>
      </fo:list-item-label>
      <fo:list-item-body start-indent="body-start()">
        <fo:block color="blue">92</fo:block>
      </fo:list-item-body>
    </fo:list-item>
  </fo:list-block>

```

下圖是 ScoreCard.xml 的 XSL-FO 範本 ScoreCard.fo。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <fo:layout-master-set>
4     <fo:simple-page-master master-name="Card" page-height="8cm"
5       page-width="12cm" margin="0.5cm 0.5cm 0.5cm 0.5cm">
6       <fo:region-body margin-top="0.5cm" margin-bottom="0.8cm"/>
7       <fo:region-before extent="1cm"/>
8     </fo:simple-page-master>
9   </fo:layout-master-set>
10  <fo:page-sequence master-reference="Card">
11    <fo:static-content flow-name="xsl-region-before">
12      <fo:block text-align="center" font-size="10pt"
13        color="blue">
14        <fo:inline text-decoration="underline">
15          Class Score Report Card
16        </fo:inline>
17      </fo:block>
18    </fo:static-content>
19    <fo:flow flow-name="xsl-region-body">
20      <fo:block font-size="12pt">Report
21      Date:2002-08-21</fo:block>
22      <fo:block font-size="12pt">Student Name:Mahler</fo:block>
23      <fo:block font-size="12pt">Student ID:41003</fo:block>
24      <fo:block font-size="12pt">Class ID: (Science
25      Class)</fo:block>
26      <fo:block space-before="0.3cm" text-align="center">
27      -----Score List -----
28    </fo:block>
29    <fo:list-block space-before="0.2cm"
30      provisional-distance-between-starts="3cm"
31      provisional-label-separation="0.4cm"
32      start-indent="1cm">
33      <fo:list-item>
34        <fo:list-item-label>
35          <fo:block>English :</fo:block>
36        </fo:list-item-label>
37        <fo:list-item-body start-indent="body-start()">
38          <fo:block color="blue">97</fo:block>
39        </fo:list-item-body>
40      </fo:list-item>

```

```

37         <fo:list-item>
38             <fo:list-item-label>
39                 <fo:block>Math :</fo:block>
40             </fo:list-item-label>
41             <fo:list-item-body start-indent="body-start()">
42                 <fo:block color="blue">68</fo:block>
43             </fo:list-item-body>
44         </fo:list-item>
45         <fo:list-item>
46             <fo:list-item-label>
47                 <fo:block>Science :</fo:block>
48             </fo:list-item-label>
49             <fo:list-item-body start-indent="body-start()">
50                 <fo:block color="blue">86</fo:block>
51             </fo:list-item-body>
52         </fo:list-item>
53         <fo:list-item>
54             <fo:list-item-label>
55                 <fo:block>Physics :</fo:block>
56             </fo:list-item-label>
57             <fo:list-item-body start-indent="body-start()">
58                 <fo:block color="blue">75</fo:block>
59             </fo:list-item-body>
60         </fo:list-item>
61         <fo:list-item>
62             <fo:list-item-label>
63                 <fo:block>Biology :</fo:block>
64             </fo:list-item-label>
65             <fo:list-item-body start-indent="body-start()">
66                 <fo:block color="blue">92</fo:block>
67             </fo:list-item-body>
68         </fo:list-item>
69     </fo:list-block>
70 </fo:flow>
71 </fo:page-sequence>
72 </fo:root>

```

圖 7.35 ScoreCard.fo

第四個步驟、設計樣式表

設計樣式表時，首先將頁面資訊置放在處理根元素的樣版(Template)之中，然後不同元素的處理範本中加入 XSL-FO 定義。下面是根元素的處理範本：

```

<xsl:template match="/">
  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
    <fo:layout-master-set>
      <fo:simple-page-master master-name="Card"
        page-height="8cm" page-width="12cm"
        margin="0.5cm 0.5cm 0.5cm 0.5cm">
        <fo:region-body margin-top="0.5cm"
margin-bottom="0.8cm"/>
        <fo:region-before extent="1cm"/>
      </fo:simple-page-master>
    </fo:layout-master-set>
    <fo:page-sequence master-reference="Card">
      <fo:static-content flow-name="xsl-region-before">
        <fo:block text-align="center" font-size="10pt"
color="blue">
          <fo:inline text-decoration="underline">
            Class Score Report Card

```

```

        </fo:inline>
      </fo:block>
    </fo:static-content>
    <fo:flow flow-name="xsl-region-body">
      <xsl:apply-templates/>
    </fo:flow>
  </fo:page-sequence>
</fo:root>
</xsl:template>

```

在根元素的處理範本中主要是生成 XSL-FO 基本元素，例如 <fo:root> 與 <fo:layout-master-set> 和部份 <fo:page-sequence>。在 <fo:page-sequence> 中也連帶將頁首資訊也一併生成，最後呼叫 <xsl:apply-templates> 關鍵位置生成就是頁面的內容之所在，在此 XSLT 會尋找 ScoreCard.xml 中其他元素的處理樣版來產生結果樹。

下面是處理學生基本資訊的元素，文科班與理科班學生的處理方式一樣，只是在產生 XSL-FO 時，在班別欄位的描述上多了辨別兩班的文字。所以在此僅介紹處理理科專班元素的範本。

```

<xsl:template match="ScoreCardInScienceClass">
  <fo:block font-size="12pt">Report Date:<xsl:value-of select="@Date"/>
</fo:block>
  <fo:block font-size="12pt">Student Name:<xsl:value-of select="@Name"/>
</fo:block>
  <fo:block font-size="12pt">Student ID:<xsl:value-of
select="@StudentId"/>
</fo:block>
  <fo:block font-size="12pt">Class ID:<xsl:value-of select="@ClassID"/>
  (Science Class)
</fo:block>
  <fo:block space-before="0.3cm" text-align="center">
-----Score List -----
</fo:block>
  <fo:list-block space-before="0.2cm"
provisional-distance-between-starts="3cm"
provisional-label-separation="0.4cm" start-indent="1cm">
  <xsl:for-each select="*">
    <fo:list-item>
      <fo:list-item-label>
        <fo:block>
          <xsl:value-of select="name(.)"/> :</fo:block>
        </fo:list-item-label>
        <fo:list-item-body start-indent="body-start()">
          <xsl:if test="number(.) >= 60">
            <fo:block color="blue">
              <xsl:value-of select="."/>
            </fo:block>
          </xsl:if>
          <xsl:if test="number(.) < 60">
            <fo:block color="red">
              <xsl:value-of select="."/>
            </fo:block>
          </xsl:if>
        </fo:list-item-body>
      </fo:list-item>
    </xsl:for-each>
  </fo:list-block>
</xsl:template>

```

上面的處理範本比較特別需要介紹的部份是位於處理學生成績的清單項目 <fo:list-item-body> 之中，在這個需要使用 <xsl:if> 來辨別學生的成績，如果大

於六十分就採用顏色為 color 屬性為藍色的 <fo:block> 否則就採用 color 屬性為紅色的 <fo:block>。

完整的 XSLT 樣式表如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   xmlns:fo="http://www.w3.org/1999/XSL/Format">
4   <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
5   <xsl:template match="/">
6     <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
7       <fo:layout-master-set>
8         <fo:simple-page-master master-name="Card" page-height="8cm"
9           page-width="12cm" margin="0.5cm 0.5cm 0.5cm 0.5cm">
10          <fo:region-body margin-top="0.5cm" margin-bottom="0.8cm"/>
11          <fo:region-before extent="1cm"/>
12        </fo:simple-page-master>
13      </fo:layout-master-set>
14      <fo:page-sequence master-reference="Card">
15        <fo:static-content flow-name="xsl-region-before">
16          <fo:block text-align="center" font-size="10pt" color="blue">
17            <fo:inline text-decoration="underline">
18              Class Score Report Card
19            </fo:inline>
20          </fo:block>
21        </fo:static-content>
22        <fo:flow flow-name="xsl-region-body">
23          <xsl:apply-templates/>
24        </fo:flow>
25      </fo:page-sequence>
26    </fo:root>
27  </xsl:template>
28  <xsl:template match="ScoreCardInScienceClass">
29    <fo:block font-size="12pt">Report Date:<xsl:value-of select="@Date"/>
30  </fo:block>
31  <fo:block font-size="12pt">Student Name:<xsl:value-of select="@Name"/>
32  </fo:block>
33  <fo:block font-size="12pt">Student ID:<xsl:value-of select="@StudentId"/>
34  </fo:block>
35  <fo:block font-size="12pt">Class ID:<xsl:value-of select="@ClassID"/>
36    (Science Class)
37  </fo:block>
38  <fo:block space-before="0.3cm" text-align="center">
39    -----Score List -----
40  </fo:block>
41  <fo:list-block space-before="0.2cm"
42    provisional-distance-between-starts="3cm"
43    provisional-label-separation="0.4cm" start-indent="1cm">
44    <xsl:for-each select="*">
45      <fo:list-item>
46        <fo:list-item-label>
47          <fo:block>
48            <xsl:value-of select="name(.)"/> :</fo:block>
49        </fo:list-item-label>
50        <fo:list-item-body start-indent="body-start()">
51          <xsl:if test="number(.) >= 60">
52            <fo:block color="blue">
53              <xsl:value-of select="."/>
54            </fo:block>
55          </xsl:if>
56          <xsl:if test="number(.) < 60">
57            <fo:block color="red">
```

```

58         <xsl:value-of select="."/>
59     </fo:block>
60 </xsl:if>
61 </fo:list-item-body>
62 </fo:list-item>
63 </xsl:for-each>
64 </fo:list-block>
65 </xsl:template>
66 <xsl:template match="ScoreCardInClassicClass">
67     <fo:block font-size="12pt">Report Date:<xsl:value-of select="@Date"/>
68 </fo:block>
69     <fo:block font-size="12pt">Student Name:<xsl:value-of select="@Name"/>
70 </fo:block>
71     <fo:block font-size="12pt">Student ID:<xsl:value-of select="@StudentId"/>
72 </fo:block>
73     <fo:block font-size="12pt">Class ID:<xsl:value-of select="@ClassID"/>
74     (Classic Class)
75 </fo:block>
76 <fo:block space-before="0.3cm" text-align="center">
77     -----Score List -----
78 </fo:block>
79 <fo:list-block space-before="0.2cm"
80     provisional-distance-between-starts="3cm"
81     provisional-label-separation="0.4cm" start-indent="1cm">
82     <xsl:for-each select="*">
83         <fo:list-item>
84             <fo:list-item-label>
85                 <fo:block>
86                     <xsl:value-of select="name(.)"/> :</fo:block>
87                 </fo:list-item-label>
88                 <fo:list-item-body start-indent="body-start()">
89                     <xsl:if test="number(.) >= 60">
90                         <fo:block color="blue">
91                             <xsl:value-of select="."/>
92                         </fo:block>
93                     </xsl:if>
94                     <xsl:if test="number(.) < 60">
95                         <fo:block color="red">
96                             <xsl:value-of select="."/>
97                         </fo:block>
98                     </xsl:if>
99                 </fo:list-item-body>
100             </fo:list-item>
101         </xsl:for-each>
102     </fo:list-block>
103 </xsl:template>
104 </xsl:stylesheet>

```

圖 7.36 ScoreCard2Fo.xslt

第五個步驟、產生 XSL-FO ，並輸出結果

以下是 ScoreCard.xml 使用 XSLT 搭配樣式表 ScoreCard2Fo.xslt 所產生的結果。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3     <fo:layout-master-set>
4         <fo:simple-page-master master-name="Card" page-height="8cm"
5             page-width="12cm" margin="0.5cm 0.5cm 0.5cm 0.5cm">
6             <fo:region-body margin-top="0.5cm" margin-bottom="0.8cm"/>

```

```

6         <fo:region-before extent="1cm"/>
7     </fo:simple-page-master>
8 </fo:layout-master-set>
9 <fo:page-sequence master-reference="Card">
10     <fo:static-content flow-name="xsl-region-before">
11         <fo:block text-align="center" font-size="10pt" color="blue">
12             <fo:inline text-decoration="underline">Class Score Report Card</fo:inline>
13         </fo:block>
14     </fo:static-content>
15     <fo:flow flow-name="xsl-region-body">
16         <fo:block font-size="12pt">Report Date:2002-08-21</fo:block>
17         <fo:block font-size="12pt">Student Name:Mahler</fo:block>
18         <fo:block font-size="12pt">Student ID:41003</fo:block>
19         <fo:block font-size="12pt">Class ID: (Science Class)</fo:block>
20         <fo:block space-before="0.3cm" text-align="center">
21             -----Score List -----
22         </fo:block>
23         <fo:list-block space-before="0.2cm" provisional-distance-between-starts="3cm"
24             provisional-label-separation="0.4cm" start-indent="1cm">
25             <fo:list-item>
26                 <fo:list-item-label>
27                     <fo:block>English :</fo:block>
28                 </fo:list-item-label>
29                 <fo:list-item-body start-indent="body-start()">
30                     <fo:block color="blue">97</fo:block>
31                 </fo:list-item-body>
32             </fo:list-item>
33             <fo:list-item>
34                 <fo:list-item-label>
35                     <fo:block>Math :</fo:block>
36                 </fo:list-item-label>
37                 <fo:list-item-body start-indent="body-start()">
38                     <fo:block color="blue">68</fo:block>
39                 </fo:list-item-body>
40             </fo:list-item>
41             <fo:list-item>
42                 <fo:list-item-label>
43                     <fo:block>Science :</fo:block>
44                 </fo:list-item-label>
45                 <fo:list-item-body start-indent="body-start()">
46                     <fo:block color="blue">86</fo:block>
47                 </fo:list-item-body>
48             </fo:list-item>
49             <fo:list-item>
50                 <fo:list-item-label>
51                     <fo:block>Physics :</fo:block>
52                 </fo:list-item-label>
53                 <fo:list-item-body start-indent="body-start()">
54                     <fo:block color="blue">75</fo:block>
55                 </fo:list-item-body>
56             </fo:list-item>
57             <fo:list-item>
58                 <fo:list-item-label>
59                     <fo:block>Biology :</fo:block>
60                 </fo:list-item-label>
61                 <fo:list-item-body start-indent="body-start()">
62                     <fo:block color="blue">92</fo:block>
63                 </fo:list-item-body>
64             </fo:list-item>
65         </fo:list-block>
66         <fo:block font-size="12pt">Report Date:2002-08-21</fo:block>
67         <fo:block font-size="12pt">Student Name:Jason</fo:block>
68         <fo:block font-size="12pt">Student ID:42001</fo:block>
69         <fo:block font-size="12pt">Class ID: (Classic Class)</fo:block>
70         <fo:block space-before="0.3cm" text-align="center">
71             -----Score List -----
72         </fo:block>
73         <fo:list-block space-before="0.2cm" provisional-distance-between-starts="3cm"
74             provisional-label-separation="0.4cm" start-indent="1cm">

```

```

69      <fo:list-item>
70          <fo:list-item-label>
71              <fo:block>English :</fo:block>
72          </fo:list-item-label>
73          <fo:list-item-body start-indent="body-start()">
74              <fo:block color="red">56</fo:block>
75          </fo:list-item-body>
76      </fo:list-item>
77      <fo:list-item>
78          <fo:list-item-label>
79              <fo:block>Math :</fo:block>
80          </fo:list-item-label>
81          <fo:list-item-body start-indent="body-start()">
82              <fo:block color="blue">61</fo:block>
83          </fo:list-item-body>
84      </fo:list-item>
85      <fo:list-item>
86          <fo:list-item-label>
87              <fo:block>Sociology :</fo:block>
88          </fo:list-item-label>
89          <fo:list-item-body start-indent="body-start()">
90              <fo:block color="blue">82</fo:block>
91          </fo:list-item-body>
92      </fo:list-item>
93      <fo:list-item>
94          <fo:list-item-label>
95              <fo:block>History :</fo:block>
96          </fo:list-item-label>
97          <fo:list-item-body start-indent="body-start()">
98              <fo:block color="blue">92</fo:block>
99          </fo:list-item-body>
100     </fo:list-item>
101     <fo:list-item>
102         <fo:list-item-label>
103             <fo:block>Music :</fo:block>
104         </fo:list-item-label>
105         <fo:list-item-body start-indent="body-start()">
106             <fo:block color="blue">97</fo:block>
107         </fo:list-item-body>
108     </fo:list-item>
109 </fo:list-block>
110 <fo:block font-size="12pt">Report Date:2002-08-20</fo:block>
111 <fo:block font-size="12pt">Student Name:Peter</fo:block>
112 <fo:block font-size="12pt">Student ID:41004</fo:block>
113 <fo:block font-size="12pt">Class ID: (Science Class)</fo:block>
114 <fo:block space-before="0.3cm" text-align="center">
-----Score List -----
</fo:block>
115 <fo:list-block space-before="0.2cm" provisional-distance-between-starts="3cm"
provisional-label-separation="0.4cm" start-indent="1cm">
<fo:list-item>
116     <fo:list-item-label>
117         <fo:block>English :</fo:block>
118     </fo:list-item-label>
119     <fo:list-item-body start-indent="body-start()">
120         <fo:block color="blue">67</fo:block>
121     </fo:list-item-body>
122 </fo:list-item>
123 <fo:list-item>
124     <fo:list-item-label>
125         <fo:block>Math :</fo:block>
126     </fo:list-item-label>
127     <fo:list-item-body start-indent="body-start()">
128         <fo:block color="red">34</fo:block>
129     </fo:list-item-body>
130 </fo:list-item>
131 <fo:list-item>
132     <fo:list-item-label>
133         <fo:block>Science :</fo:block>
134     </fo:list-item-label>

```

```

135         </fo:list-item-label>
136         <fo:list-item-body start-indent="body-start()">
137             <fo:block color="blue">66</fo:block>
138         </fo:list-item-body>
139     </fo:list-item>
140     <fo:list-item>
141         <fo:list-item-label>
142             <fo:block>Physics :</fo:block>
143         </fo:list-item-label>
144         <fo:list-item-body start-indent="body-start()">
145             <fo:block color="red">33</fo:block>
146         </fo:list-item-body>
147     </fo:list-item>
148     <fo:list-item>
149         <fo:list-item-label>
150             <fo:block>Biology :</fo:block>
151         </fo:list-item-label>
152         <fo:list-item-body start-indent="body-start()">
153             <fo:block color="red">54</fo:block>
154         </fo:list-item-body>
155     </fo:list-item>
156 </fo:list-block>
157 <fo:block font-size="12pt">Report Date:2002-04-13</fo:block>
158 <fo:block font-size="12pt">Student Name:Jessica</fo:block>
159 <fo:block font-size="12pt">Student ID:42004</fo:block>
160 <fo:block font-size="12pt">Class ID: (Classic Class)</fo:block>
161 <fo:block space-before="0.3cm" text-align="center">
-----Score List -----
</fo:block>
162 <fo:list-block space-before="0.2cm" provisional-distance-between-starts="3cm"
provisional-label-separation="0.4cm" start-indent="1cm">
<fo:list-item>
163     <fo:list-item-label>
164         <fo:block>English :</fo:block>
165     </fo:list-item-label>
166     <fo:list-item-body start-indent="body-start()">
167         <fo:block color="blue">100</fo:block>
168     </fo:list-item-body>
169 </fo:list-item>
170 <fo:list-item>
171     <fo:list-item-label>
172         <fo:block>Math :</fo:block>
173     </fo:list-item-label>
174     <fo:list-item-body start-indent="body-start()">
175         <fo:block color="red">54</fo:block>
176     </fo:list-item-body>
177 </fo:list-item>
178 <fo:list-item>
179     <fo:list-item-label>
180         <fo:block>Sociology :</fo:block>
181     </fo:list-item-label>
182     <fo:list-item-body start-indent="body-start()">
183         <fo:block color="red">44</fo:block>
184     </fo:list-item-body>
185 </fo:list-item>
186 <fo:list-item>
187     <fo:list-item-label>
188         <fo:block>History :</fo:block>
189     </fo:list-item-label>
190     <fo:list-item-body start-indent="body-start()">
191         <fo:block color="red">59</fo:block>
192     </fo:list-item-body>
193 </fo:list-item>
194 <fo:list-item>
195     <fo:list-item-label>
196         <fo:block>Music :</fo:block>
197     </fo:list-item-label>
198     <fo:list-item-body start-indent="body-start()">
199         <fo:block color="blue">100</fo:block>
200

```

```

201 |                                     </fo:list-item-body>
202 |                                 </fo:list-item>
203 |                             </fo:list-block>
204 |                         </fo:flow>
205 |                     </fo:page-sequence>
206 | </fo:root>

```

圖 7.37 ScoreCard.xml 使用 ScoreCard2Fo.xslt 產生結果

下圖為圖 7.37 經由 XSL-FO 處理器所產生之 PDF 畫面，一共四頁。

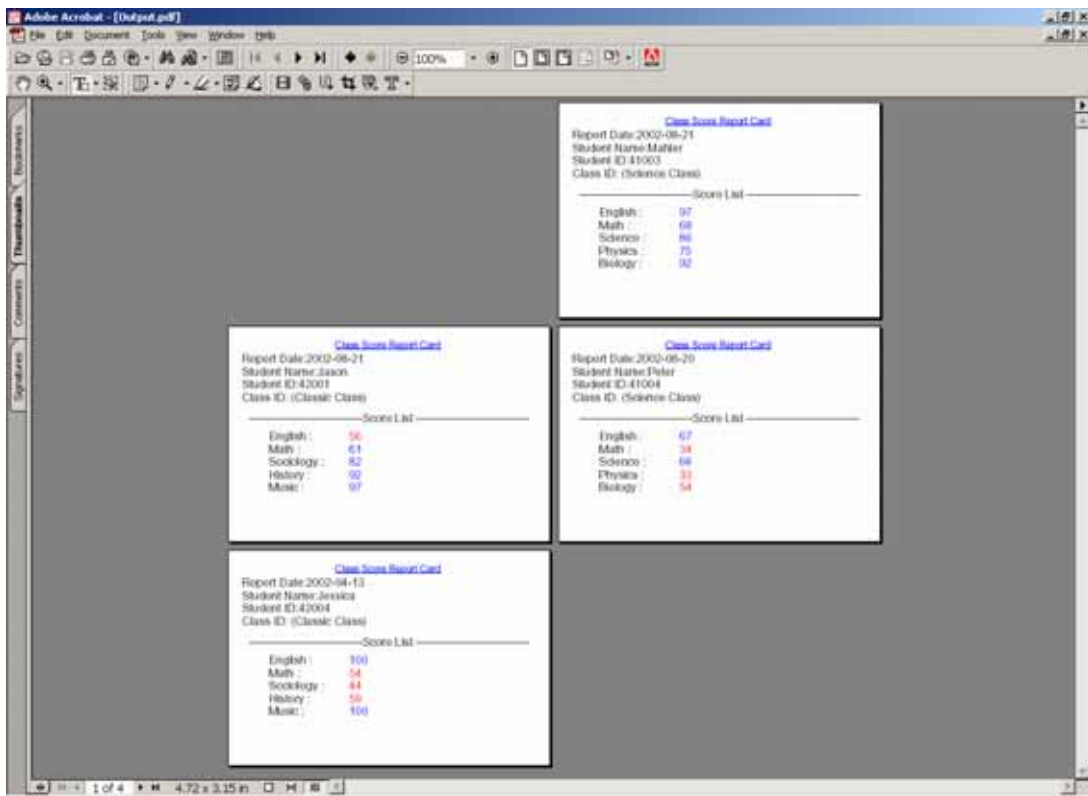


圖 7.38 ScoreCard.xml 輸出 PDF 結果

7.5.2 由 HTML 產生 XSL-FO

7.5.2.1 處理 <head>, <body> 元素

<head> 與 <body> 元素都是根元素 <html> 的子元素，<head> 元素中有一子元素 <title> 用來標記文件的標題，而 <body> 元素的內容就是網頁文件的內容。下圖是一個簡單的 HTML 範例 Hello.html 將用來討論 HTML 文件如何使用 XSLT 建立 XSL-FO 文件。

```

1 | <html>
2 |   <head>
3 |     <title>Hello World Sample</title>

```

```

4     </head>
5     <body><p>Hello XSL-Transformation and XSL-Formatting Objects</p>
6     </body>
7 </html>

```

圖 7.40 Hello.html

在 Hello.html 中，<title> 元素可以用來作為輸出文件的封面或是頁尾，而<body> 元素的內容因為元素種類眾多，在後續將會針對不同 HTML 元素，例如表格、清單等等的處理進行詳盡的介紹。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:fo="http://www.w3.org/1999/XSL/Format">
5   <xsl:output method="xml" version="1.0" encoding="UTF-8"
6     indent="yes"/>
7   <xsl:template match="/">
8     <xsl:apply-templates/>
9   </xsl:template>
10  <xsl:template match="head"/>
11  <xsl:template match="body">
12    <!-- Create Page Master -->
13    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
14      <fo:layout-master-set>
15        <fo:simple-page-master master-name="A4"
16          page-height="29.8cm" page-width="21cm"
17          margin="1.5cm 1.5cm 1.5 1.5cm">
18          <fo:region-body margin-top="1cm"
19            margin-bottom="1cm"/>
20          <fo:region-after extent="1cm"/>
21        </fo:simple-page-master>
22      </fo:layout-master-set>
23      <fo:page-sequence master-reference="A4">
24        <!-- Create The Footer of Page -->
25        <fo:static-content flow-name="xsl-region-after">
26          <fo:block>
27            <xsl:value-of select="/html/head/title"/>
28          </fo:block>
29        </fo:static-content>
30        <!-- Create The Content of Page -->
31        <fo:flow flow-name="xsl-region-body">
32          <!-- Process Other Child Element in Body -->
33          <xsl:apply-templates/>
34        </fo:flow>
35      </fo:page-sequence>
36    </fo:root>
37  </xsl:template>
38  <xsl:template match="p">
39    <fo:block>
40      <xsl:value-of select="."/>
41    </fo:block>
42  </xsl:template>
43 </xsl:stylesheet>

```

圖 7.41 Hello.xslt

圖 7.41 的第 9 到 34 行是處理 <body> 元素的樣版，在這個樣版中必須建立好基本的 XSL-FO 文件架構。這個架構必須是由 <fo:root>、<fo:layout-master-set>、<fo:page-sequence> 等元素所構成，而且要將各個元素的屬性值一併填入，以 <fo:simple-pagemaster> 來說，頁面大小、邊界等等屬性

可以直接代入。

第 22 到 26 行是將 `<title>` 元素內容取得作為輸出頁面的頁尾，而第 30 行是處理其他 `<body>` 的內容，透過呼叫 `<xsl:apply-templates/>` 尋找適合處理的樣版產生適當的 XSL-FO 結果樹。

下圖是 Hello.html 使用樣式表 Hello.xslt 所產生的 XSL-FO 結果與 PDF 輸出畫面。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <fo:layout-master-set>
4     <fo:simple-page-master master-name="A4"
5       page-height="29.8cm" page-width="21cm"
6       margin="1.5cm 1.5cm 1.5 1.5cm">
7       <fo:region-body margin-top="1cm" margin-bottom="1cm"/>
8       <fo:region-after extent="1cm"/>
9     </fo:simple-page-master>
10  </fo:layout-master-set>
11  <fo:page-sequence master-reference="A4">
12    <fo:static-content flow-name="xsl-region-after">
13      <fo:block>Hello World Sample</fo:block>
14    </fo:static-content>
15    <fo:flow flow-name="xsl-region-body">
16      <fo:block>
17        Hello XSL-Transformation and XSL-Formatting Objects
18      </fo:block>
19    </fo:flow>
20  </fo:page-sequence>
21 </fo:root>
```

圖 7.42 Hello.html 搭配樣式表的輸出結果

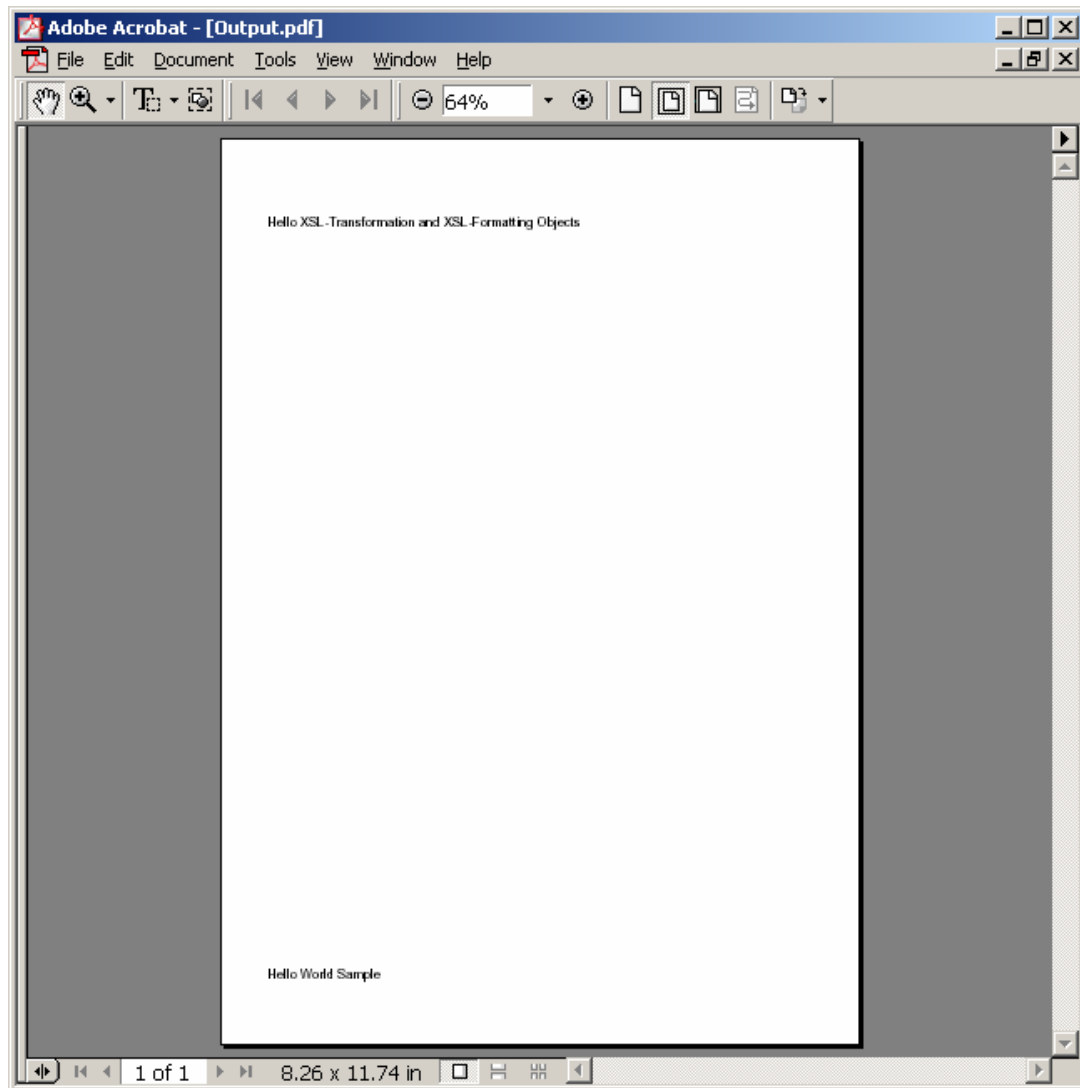


圖 7.43 Hello.html 輸出 PDF 結果畫面

7.5.2.2 處理標題元素

HTML4 中提供了六種標記標題(Heading)的元素，這些元素分別是 `<h1>`，`<h2>`，`<h3>`，`<h4>`，`<h5>`，`<h6>`，下圖 7.44 就是標題元素的範例：

```
1 <?xml-stylesheet type="text/xsl" href="Heading.xslt"?>
2 <html>
3   <head>
4     <title>Process Heading Element</title>
5   </head>
6   <body>
7     <h1>This is the title of &lt;h1&gt;</h1>
8     <h2>This is the title of &lt;h2&gt;</h2>
9     <h3>This is the title of &lt;h3&gt;</h3>
10    <h4>This is the title of &lt;h4&gt;</h4>
11    <h5>This is the title of &lt;h5&gt;</h5>
12    <h6>This is the title of &lt;h6&gt;</h6>
```

```

13 |         </body>
14 |     </html>

```

圖 7.44 Heading.html

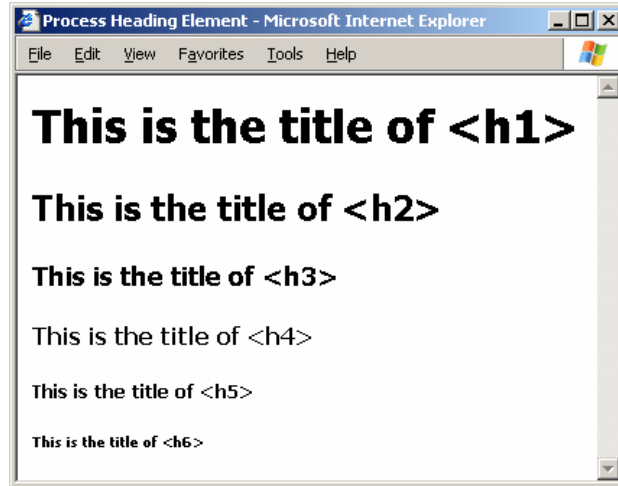


圖 7.45 Heading.html 在瀏覽器的畫面

標題元素在 XSL-FO 的角度來看，就是一種 `<fo:block>`，只是六種不同的標題元素，需要建立不同的 `<fo:block>`，例如標題一的字體大小要比標題二大，或是標題一需要由新的一頁開始等等。下表依據六種標題元素建立了不同 `<fo:block>` 樣式。

元素	屬性集合名稱	<code><fo:block></code> 樣式
<code><h1></code>	heading1	字體大小 26pt, 字型 sans-serif, Bold, space-after: 16pt, break-before: page (換頁)
<code><h2></code>	heading2	字體大小 22pt, 字型 sans-serif, Bold, space-before: 20pt, space-after: 6pt
<code><h3></code>	heading3	字體大小 17pt, 字型 sans-serif, Bold, space-before: 12pt, space-after: 5pt
<code><h4></code>	heading4	字體大小 14pt, 字型 sans-serif, Bold, space-before: 8pt, space-after: 5pt
<code><h5></code>	heading5	字體大小 12pt, 字型 sans-serif, Bold, space-before: 5pt, space-after: 5pt
<code><h6></code>	heading6	字體大小 10pt, 字型 sans-serif, Bold, space-before: 5pt, space-after: 3pt

圖 7.46 標題元素樣式描述

在標題元素中除了 `<h1>` 是有換頁需求以外，其他標題元素的樣式需求都是差不多的，例如字體大小、字型、與前後物件的距離等等。一旦樣式描述建立完成後，接下來建立 XSLT 的樣式表。為了使樣式表看起來簡潔，將使用 XSLT 的 `<attribute-set>` 元素來將上述六種標題元素的 `<fo:block>` 屬性集合起來，並

由 heading1 至 heading6 命名。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4     xmlns:fo="http://www.w3.org/1999/XSL/Format">
5   <xsl:output method="xml" version="1.0" encoding="UTF-8"
6     indent="yes"/>
7
8   <!-- Define Heading Element fo:block Style Attribute -->
9   <xsl:attribute-set name="heading1">
10    <xsl:attribute name="font-size">26pt</xsl:attribute>
11    <xsl:attribute name="font-family">sans-serif</xsl:attribute>
12    <xsl:attribute name="font-weight">bold</xsl:attribute>
13    <xsl:attribute name="space-after">16pt</xsl:attribute>
14    <xsl:attribute name="break-before">page</xsl:attribute>
15  </xsl:attribute-set>
16  <xsl:attribute-set name="heading2">
17    <xsl:attribute name="font-size">22pt</xsl:attribute>
18    <xsl:attribute name="font-family">sans-serif</xsl:attribute>
19    <xsl:attribute name="font-weight">bold</xsl:attribute>
20    <xsl:attribute name="space-before">20pt</xsl:attribute>
21    <xsl:attribute name="space-after">6pt</xsl:attribute>
22  </xsl:attribute-set>
23  <xsl:attribute-set name="heading3">
24    <xsl:attribute name="font-size">17pt</xsl:attribute>
25    <xsl:attribute name="font-family">sans-serif</xsl:attribute>
26    <xsl:attribute name="font-weight">bold</xsl:attribute>
27    <xsl:attribute name="space-before">12pt</xsl:attribute>
28    <xsl:attribute name="space-after">5pt</xsl:attribute>
29  </xsl:attribute-set>
30  <xsl:attribute-set name="heading4">
31    <xsl:attribute name="font-size">14pt</xsl:attribute>
32    <xsl:attribute name="font-family">sans-serif</xsl:attribute>
33    <xsl:attribute name="font-weight">bold</xsl:attribute>
34    <xsl:attribute name="space-before">8pt</xsl:attribute>
35    <xsl:attribute name="space-after">5pt</xsl:attribute>
36  </xsl:attribute-set>
37  <xsl:attribute-set name="heading5">
38    <xsl:attribute name="font-size">12pt</xsl:attribute>
39    <xsl:attribute name="font-family">sans-serif</xsl:attribute>
40    <xsl:attribute name="font-weight">bold</xsl:attribute>
41    <xsl:attribute name="space-before">5pt</xsl:attribute>
42    <xsl:attribute name="space-after">5pt</xsl:attribute>
43  </xsl:attribute-set>
44  <xsl:attribute-set name="heading6">
45    <xsl:attribute name="font-size">10pt</xsl:attribute>
46    <xsl:attribute name="font-family">sans-serif</xsl:attribute>
47    <xsl:attribute name="font-weight">bold</xsl:attribute>
48    <xsl:attribute name="space-before">5pt</xsl:attribute>
49    <xsl:attribute name="space-after">3pt</xsl:attribute>
50  </xsl:attribute-set>
51  <xsl:template match="/">
52    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
53      <fo:layout-master-set>
54        <fo:simple-page-master master-name="A4"
55          page-height="29.8cm"
56          page-width="21cm" margin="1.5cm 1.5cm 1.5cm 1.5cm">
57          <fo:region-body margin-top="1cm"
58            margin-bottom="1cm"/>
59        </fo:simple-page-master>
60      </fo:layout-master-set>
```

```

58         <fo:page-sequence master-reference="A4">
59             <fo:flow flow-name="xsl-region-body">
60                 <xsl:apply-templates/>
61             </fo:flow>
62         </fo:page-sequence>
63     </fo:root>
64 </xsl:template>
65 <xsl:template match="h1">
66     <fo:block xsl:use-attribute-sets="heading1" >
67         <xsl:value-of select="."/></fo:block>
68 </xsl:template>
69 <xsl:template match="h2">
70     <fo:block xsl:use-attribute-sets="heading2" >
71         <xsl:value-of select="."/></fo:block>
72 </xsl:template>
73 <xsl:template match="h3">
74     <fo:block xsl:use-attribute-sets="heading3" >
75         <xsl:value-of select="."/></fo:block>
76 </xsl:template>
77 <xsl:template match="h4">
78     <fo:block xsl:use-attribute-sets="heading4" >
79         <xsl:value-of select="."/></fo:block>
80 </xsl:template>
81 <xsl:template match="h5">
82     <fo:block xsl:use-attribute-sets="heading5" >
83         <xsl:value-of select="."/></fo:block>
84 </xsl:template>
85 <xsl:template match="h6">
86     <fo:block xsl:use-attribute-sets="heading6" >
87         <xsl:value-of select="."/></fo:block>
88 </xsl:template>
89 </xsl:stylesheet>

```

圖 7.47 Heading.xslt

透過 Heading.xslt 所產生出來的 <h1> 元素結果為：

```

<fo:block font-size="26pt"
          font-family="sans-serif"
          font-weight="bold"
          space-after="16pt"
          break-before="page">
    This is the title of &lt;h1&gt;
</fo:block>

```

Heading.html 透過樣式表轉換後使用 XSL-FO 處理器的輸出 PDF 的結果如下。

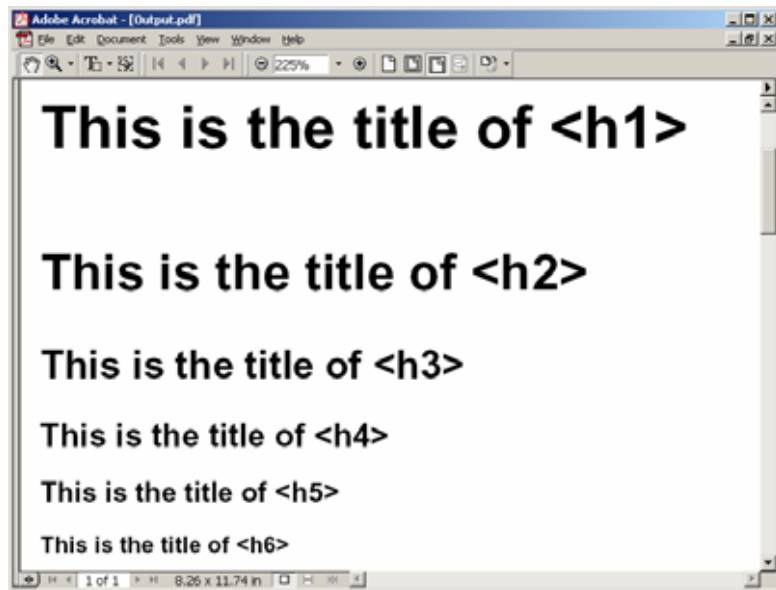


圖 7.48 Heading.html 輸出成 PDF 結果畫面

7.5.2.3 處理 , <i>, <u> 元素

元素 , <i>, <u> 分別是將標記的內容以粗體(Bold)、斜體(Italic)、下加底線(Underline)的方式呈現，下面範例 Hello_Style.html 是修改來自 Hello.html，將原本的網頁內容做粗體、斜體、底線等等應用。

```
1 <?xml-stylesheet type="text/xsl" href="Hello.xslt"?>
2 <html>
3   <head>
4     <title>Hello World Sample</title>
5   </head>
6   <body>
7     <p>
8       <b>Hello</b>
9       <i>XSL-Transformation</i> and <u>XSL-Formatting Objects</u>
10    </p>
11  </body>
12 </html>
```

圖 7.49 Hello_Style.html

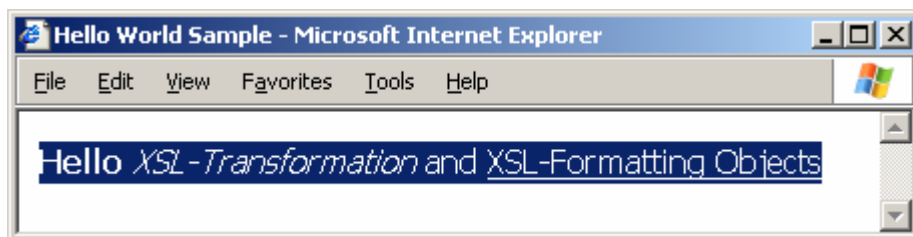


圖 7.49 Hello_Style.html 在瀏覽器中呈現的結果

由於 ``, `<i>`, `<u>` 等諸如此類的臨時樣式處理標記都是用於段落中的關鍵字或是重要句子，用以加強閱讀時的視覺效果，所以 XSL-FO 為這一類的元素設計了 `<fo:inline>` 元素，`<fo:inline>` 就是為了處理這一些臨時樣式而發展的。

下面的樣式表 `Hello_Style.xslt` 是修改自 `Hello.xslt`，主要是新增了三個處理 ``, `<i>`, `<u>` 元素的樣版。

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:fo="http://www.w3.org/1999/XSL/Format">
5   <xsl:output method="xml" version="1.0" encoding="UTF-8"
6     indent="yes"/>
7   <xsl:template match="/">
8     <xsl:apply-templates/>
9   </xsl:template>
10  <xsl:template match="head"/>
11  <xsl:template match="body">
12    <!-- Create Page Master -->
13    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
14      <fo:layout-master-set>
15        <fo:simple-page-master master-name="A4"
16          page-height="29.8cm" page-width="21cm"
17          margin="1.5cm 1.5cm 1.5cm 1.5cm">
18          <fo:region-body margin-top="1cm"
19            margin-bottom="1cm"/>
20          <fo:region-after extent="1cm"/>
21        </fo:simple-page-master>
22      </fo:layout-master-set>
23      <fo:page-sequence master-reference="A4">
24        <!-- Create The Footer of Page -->
25        <fo:static-content flow-name="xsl-region-after">
26          <fo:block>
27            <xsl:value-of select="/html/head/title"/>
28          </fo:block>
29        </fo:static-content>
30        <!-- Create The Content of Page -->
31        <fo:flow flow-name="xsl-region-body">
32          <!-- Process Other Child Element in Body -->
33          <xsl:apply-templates/>
34        </fo:flow>
35      </fo:page-sequence>
36    </fo:root>
37  </xsl:template>
38  <xsl:template match="p">
39    <fo:block>
40      <xsl:apply-templates/>
41    </fo:block>
42  </xsl:template>
43  <xsl:template match="b">
44    <fo:inline font-weight="bold">
45      <xsl:apply-templates/>
46    </fo:inline>
47  </xsl:template>
48  <xsl:template match="i">
49    <fo:inline font-style="italic">
50      <xsl:apply-templates/>
51    </fo:inline>
52  </xsl:template>
```

```

50     <xsl:template match="u">
51         <fo:inline text-decoration="underline">
52             <xsl:apply-templates/>
53         </fo:inline>
54     </xsl:template>
55 </xsl:stylesheet>

```

圖 7.50 Hello_Style.xslt

在 Hello_Style.xslt 第 40 到 54 行所定義的三個樣版，這三個處理樣版皆是產生 <fo:inline> 元素，不過在三個樣版中的 <fo:inline> 屬性隨著不同的處理元素有所不同，例如：對於粗體 和斜體 <i> 分別使用 font-weight="bold" 與 font-style="italic" 屬性來描述 <fo:inline> 的呈現結果，而文字下加底線 <u> 則是使用 text-decoration="underline" 來控制 XSL-FO 將輸出文字下加底線。

以下是 Hello_Style.html 使用樣式表 Hello_Style.xslt 產生的 XSL-FO 結果與輸出 PDF 的畫面。

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3      <fo:layout-master-set>
4          <fo:simple-page-master master-name="A4" page-height="29.8cm"
5              page-width="21cm" margin="1.5cm 1.5cm 1.5cm 1.5cm">
6              <fo:region-body margin-top="1cm" margin-bottom="1cm"/>
7              <fo:region-after extent="1cm"/>
8          </fo:simple-page-master>
9      </fo:layout-master-set>
10     <fo:page-sequence master-reference="A4">
11         <fo:static-content flow-name="xsl-region-after">
12             <fo:block>Hello World Sample</fo:block>
13         </fo:static-content>
14         <fo:flow flow-name="xsl-region-body">
15             <fo:block>
16                 <fo:inline font-weight="bold">Hello</fo:inline>
17                 <fo:inline
18 font-style="italic">XSL-Transformation</fo:inline>
19                 and
20                 <fo:inline text-decoration="underline">
21                     XSL-Formatting Objects</fo:inline>
22             </fo:block>
23         </fo:flow>
24     </fo:page-sequence>
25 </fo:root>

```

圖 7.51 Hello_Style.html 搭配 Hello_Style.xslt 輸出結果

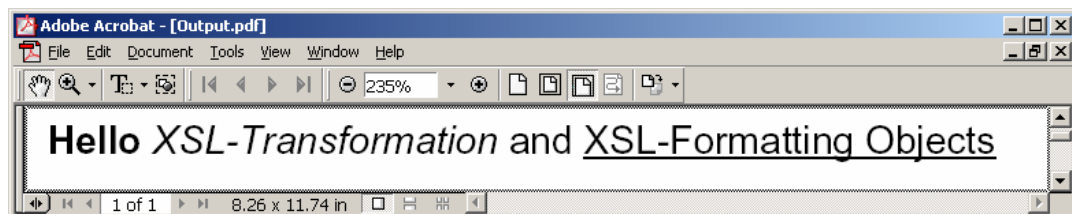


圖 7.52 PDF 輸出結果

7.5.2.4 處理 <a> 元素

由於 Anchor 元素包含了超連結 (Hypertext Reference) 屬性，而屬性的內容通常是一段 URL。因為 XSLFO 所輸出的對象大多是非互動式的媒介，例如紙張、圖形檔案等等，所以在起裡 <a> 元素時，可以將 href 屬性取出以小括號包起，例如：

W3C (<http://www.w3.org>)

下面的 HTML 範例是了示範處理 Anchor 元素撰寫了三個超連結分別連結到 Yahoo!, W3C!, 和 Google!。

```
1 <html>
2   <head>
3     <title>Processing Anchor Element</title>
4   </head>
5   <body>
6     <p><a href="http://www.yahoo.com">Yahoo!</a></p>
7     <p><a href="http://www.w3.org">W3C!</a></p>
8     <p><a href="http://www.google.com">Google!</a></p>
9   </body>
10 </html>
```

圖 7.53 Anchor.html

處理 Anchor.html 的 XSLT 樣版如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:fo="http://www.w3.org/1999/XSL/Format">
5   <xsl:output method="xml" version="1.0" encoding="UTF-8"
6     indent="yes"/>
7   <xsl:template match="/">
8     <xsl:apply-templates/>
9   </xsl:template>
10  <xsl:template match="head"/>
11  <xsl:template match="body">
12    <!-- Create Page Master -->
13    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
14      <fo:layout-master-set>
15        <fo:simple-page-master master-name="A4"
16          page-height="29.8cm"
17          page-width="21cm" margin="1.5cm 1.5cm 1.5 1.5cm">
18          <fo:region-body margin-top="1cm"
19            margin-bottom="1cm"/>
20        </fo:simple-page-master>
21      </fo:layout-master-set>
22      <fo:page-sequence master-reference="A4">
23        <fo:flow flow-name="xsl-region-body">
24          <xsl:apply-templates/>
25        </fo:flow>
26      </fo:page-sequence>
27    </fo:root>
28  </xsl:template>
29  <xsl:template match="p">
30    <fo:block>
31      <xsl:apply-templates/>
32    </fo:block>
33  </xsl:template>
34 </xsl:stylesheet>
```



```

28         </fo:block>
29     </xsl:template>
30     <xsl:template match="a">
31         <xsl:variable name="AnchorText">
32             <xsl:value-of select="."/>
33         </xsl:variable>
34         <xsl:apply-templates/>
35         <xsl:if test="@href!=$AnchorText">
36             <fo:inline>
37                 <xsl:text></xsl:text>
38                 <xsl:value-of select="@href"/>
39                 <xsl:text></xsl:text>
40             </fo:inline>
41         </xsl:if>
42     </xsl:template>
43 </xsl:stylesheet>

```

圖 7.54 Anchor.xslt

在圖 7.54 中的第 30 至 42 行是處理 Anchor 元素的樣版，此樣版首先建立變數 \$AnchorText 用來記錄 <a> 元素的內容，然後呼叫 <xsl:apply-template> 將 <a> 的內容取出，取出之後，先行測試 <a> 中的 href 屬性是否和 \$AnchorText 變數不同，如果不相同則印出由小括號所成的 href 屬性值。

Anchor.html 搭配 Anchor.xslt 產生的 XSL-FO 如下：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3     <fo:layout-master-set>
4         <fo:simple-page-master master-name="A4" page-height="29.8cm"
5             page-width="21cm" margin="1.5cm 1.5cm 1.5cm 1.5cm">
6             <fo:region-body margin-top="1cm" margin-bottom="1cm"/>
7         </fo:simple-page-master>
8     </fo:layout-master-set>
9     <fo:page-sequence master-reference="A4">
10         <fo:flow flow-name="xsl-region-body">
11             <fo:block>Yahoo!<fo:inline>(http://www.yahoo.com)</fo:inline></fo:block>
12             <fo:block>W3C!<fo:inline>(http://www.w3.org)</fo:inline></fo:block>
13             <fo:block>Google!<fo:inline>(http://www.google.com)</fo:inline></fo:block>
14         </fo:flow>
15     </fo:page-sequence>
16 </fo:root>

```

圖 7.55 Anchor.html 產生之 XSL-FO

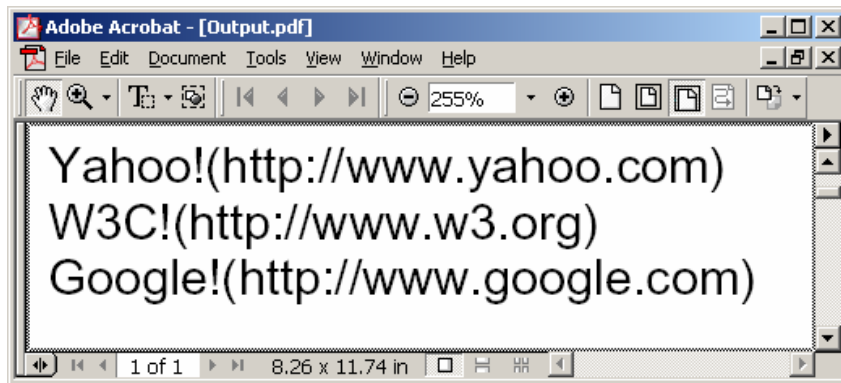


圖 7.55 PDF 輸出結果

7.5.2.5 處理 `
` 元素

處理 `
` 元素是一個空白元素，目的在於讓頁面產生換行的作用，所以要在設計 XSLT 樣版時，遇到 `
` 元素就是將其置換成爲空的 `<fo:block>` 即可。

```
<xsl:template match="br">
  <fo:blcok>
</fo:block>
</xsl:template>
```

7.5.2.6 處理 `` 元素

依據 W3C 所制訂的 HTML 4.01 規格書於 7.5.4 節中說明了 `` 元素是一種通用的 `inline` 元素，所以在設計樣式表時，當遇到 `` 元素，只要將其換成 `<fo:inline>`。由於 `` 中可能有一些 CSS 類別改變 `` 的輸出樣式，例如：字體前景爲藍色，當遇到此種情形時，就可以是情況修改處理範本中的 `<fo:inline>` 屬性。

處理 `` 的樣版如下：

```
<xsl:template match="span">
  <fo:inline>
    <xsl:apply-templates/>
  </fo:inline >
</xsl:template>
```

7.5.2.7 處理 `<p>` 元素

`<p>` 是相對於 XSL-FO 中的 `<fo:block>` 元素，也就是說，處理 `<p>` 的範本只要 `<p>` 換成 `<foblock>` 另外加上對段落樣式的屬性設定即可，下面的例子是處理 `<p>` 的 XSLT 範本。

```

<xsl:attribute-set name="Paragraph">
  <xsl:attribute name="text-indent">1.2cm</xsl:attribute>
  <xsl:attribute name="font-family">Times New Roman</xsl:attribute>
  <xsl:attribute name="font-size">10pt</xsl:attribute>
</xsl:attribute-set>
<xsl:template match="p">
  <fo:block xsl:use-attribute-sets="Paragraph">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

```

7.5.2.8 處理 Table

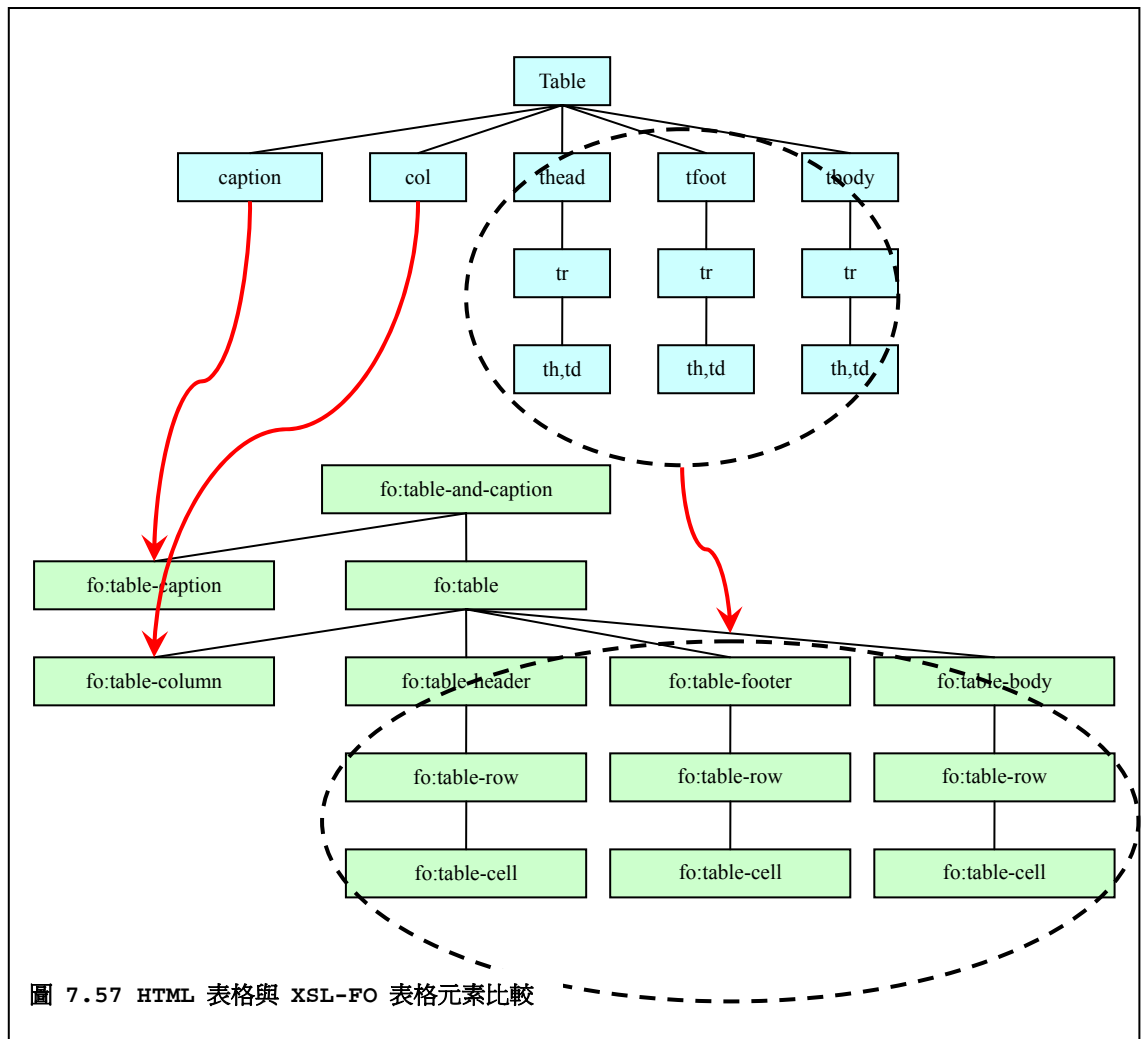
爲了要將 HTML 表格元素轉換成爲 XSL-FO 的表格元素，首先對於兩種標記語言的表格元素進行比較。

下面是 HTML 4.01 的表格元素。

元素	與 XSL-FO 對應的元素	功能
<table>	<fo:table-and-caption>	整份表格的根元素
<caption>	<fo:table-caption>	表格的標題
<col>	<fo:table-column>	表格的欄位資訊
<thead>	<fo:table-header>	標題列群組
<tfoot>	<fo:table-footer>	註腳列群組
<tbody>	<fo:table-body>	內容列群組
<tr>	<fo:table-row>	列
<th>,<td>	<fo:table-cell>	儲存格

圖 7.56 HTML 表格與 XSL 表格比較

下圖是 HTML 表格元素和 XSL-FO 表格元素相對應的圖表，透過比較的圖表會發現兩者的結構幾乎差不多，所以將 HTML 表格轉換成爲 XSL-FO 表格只需要進行素的替換而已。



下圖的 HTML 表格範例是用來測試處理 HTML 的 XSLT 範本，在這個簡單的範例中，表格是由「表格標題」、「欄位標題」、「表格列」與「儲存格」所構成，其中存放了姓名、年齡、性別三種資料。

```

1 <html>
2   <head>
3     <title>Processing Table Element</title>
4   </head>
5   <body>
6     <table border="1">
7       <caption>The Caption Of Table</caption>
8       <thead>
9         <tr>
10          <td>Name</td>
11          <td>Age</td>
12        </tr>
13      </thead>
14      <tbody>
15        <tr>
16          <td>Peter</td>
17          <td>19</td>
18        </tr>
19        <tr>
20          <td>Kevin</td>
21          <td>22</td>

```

```

22         </tr>
23     <tr>
24         <td>Wang</td>
25         <td>33</td>
26     </tr>
27     <tr>
28         <td>Helles</td>
29         <td>26</td>
30     </tr>
31 </tbody>
32 </table>
33 </body>
34 </html>

```

圖 7.58 Table.html

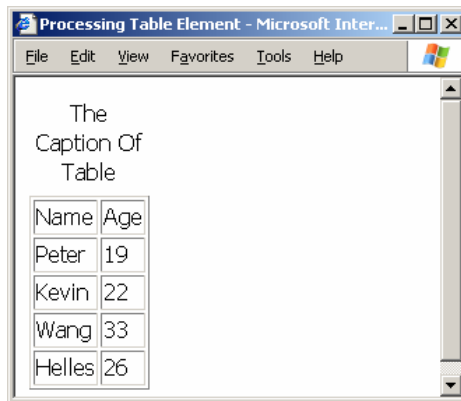


圖 7.59 Table.html 在瀏覽器中的結果

處理 Table.html 的 XSLT 樣式表中僅處理 HTML 表格元素，如果要加入其他樣式處理的範本，請參考前面適合的例子。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3     xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4     xmlns:fo="http://www.w3.org/1999/XSL/Format">
5     <xsl:output method="xml" indent="yes"/>
6     <!-- Setting Attribute Set For fo:table-caption -->
7     <xsl:attribute-set name="table-caption">
8         <xsl:attribute name="font-family">sans-serif</xsl:attribute>
9         <xsl:attribute name="text-align">center</xsl:attribute>
10        <xsl:attribute name="space-before">3pt</xsl:attribute>
11        <xsl:attribute name="space-after">3pt</xsl:attribute>
12        <xsl:attribute name="font-weight">bold</xsl:attribute>
13        <xsl:attribute
14            name="keep-with-next.within-page">always</xsl:attribute>
15    </xsl:attribute-set>
16
17    <!-- Setting Attribute Set For fo:table -->
18    <xsl:attribute-set name="table">
19        <xsl:attribute name="table-layout">fixed</xsl:attribute>
20        <xsl:attribute name="space-before">12pt</xsl:attribute>
21        <xsl:attribute name="space-after">14pt</xsl:attribute>
22    </xsl:attribute-set>
23
24    <!-- //////////////// XSLT Templates ////////////////////// -->
25    <xsl:template match="/">
26        <xsl:apply-templates/>

```

```

25     </xsl:template>
26     <xsl:template match="head" />
27     <xsl:template match="body">
28         <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
29             <fo:layout-master-set>
30                 <fo:simple-page-master master-name="A4"
31 page-height="29.8cm"
32                 page-width="21cm" margin="1.5cm 1.5cm 1.5cm 1.5cm">
33                     <fo:region-body margin-top="1cm"
34 margin-bottom="1cm" />
35                     <fo:region-after extent="1cm" />
36                 </fo:simple-page-master>
37             </fo:layout-master-set>
38             <fo:page-sequence master-reference="A4">
39                 <!-- Create The Content of Page -->
40                 <fo:flow flow-name="xsl-region-body">
41                     <!-- Process Other Child Element in Body -->
42                     <xsl:apply-templates/>
43                 </fo:flow>
44             </fo:page-sequence>
45         </fo:root>
46     </xsl:template>
47     <!-- //////////////// Table Elements Processing Templates
48     //////////////// -->
49     <xsl:template match="table">
50         <fo:table-and-caption>
51             <xsl:if test="caption">
52                 <fo:table-caption
53 xsl:use-attribute-sets="table-caption">
54                     <fo:block>
55                         <xsl:value-of select="caption" />
56                     </fo:block>
57                 </fo:table-caption>
58             </xsl:if>
59             <fo:table xsl:use-attribute-sets="table">
60                 <xsl:apply-templates/>
61             </fo:table>
62         </fo:table-and-caption>
63     </xsl:template>
64     <xsl:template match="caption" />
65     <xsl:template match="col">
66         <fo:table-column column-width="{@width}" />
67     </xsl:template>
68     <xsl:template match="thead">
69         <fo:table-header>
70             <xsl:apply-templates/>
71         </fo:table-header>
72     </xsl:template>
73     <xsl:template match="tfoot">
74         <fo:table-footer>
75             <xsl:apply-templates/>
76         </fo:table-footer>
77     </xsl:template>
78     <xsl:template match="tbody">
79         <fo:table-body>
80             <xsl:apply-templates/>
81         </fo:table-body>
82     </xsl:template>
83     <xsl:template match="tr">
84         <fo:table-row>
85             <xsl:apply-templates/>
86         </fo:table-row>
87     </xsl:template>
88     <xsl:template match="th">
89         <fo:table-cell>
90             <xsl:call-template name="cell-span" />
91             <xsl:if test="@valign">
92                 <xsl:attribute name="display-align">

```

```

89         <xsl:value-of select="@valign" />
90     </xsl:attribute>
91 </xsl:if>
92 <fo:block>
93     <xsl:if test="@align">
94         <xsl:attribute name="text-align">
95             <xsl:value-of select="@align" />
96         </xsl:attribute>
97     </xsl:if>
98     <xsl:apply-templates />
99 </fo:block>
100 </fo:table-cell>
101 </xsl:template>
102 <xsl:template match="td">
103     <fo:table-cell>
104         <xsl:call-template name="cell-span" />
105         <xsl:if test="@valign">
106             <xsl:attribute name="display-align">
107                 <xsl:value-of select="@valign" />
108             </xsl:attribute>
109         </xsl:if>
110         <fo:block>
111             <xsl:if test="@align">
112                 <xsl:attribute name="text-align">
113                     <xsl:value-of select="@align" />
114                 </xsl:attribute>
115             </xsl:if>
116             <xsl:apply-templates />
117         </fo:block>
118     </fo:table-cell>
119 </xsl:template>
120 <xsl:template name="cell-span">
121     <xsl:if test="@colspan">
122         <xsl:attribute name="number-columns-spanned">
123             <xsl:value-of select="@colspan" />
124         </xsl:attribute>
125     </xsl:if>
126     <xsl:if test="@rowspan">
127         <xsl:attribute name="number-rows-spanned">
128             <xsl:value-of select="@rowspan" />
129         </xsl:attribute>
130     </xsl:if>
131 </xsl:template>
132 </xsl:stylesheet>

```

圖 7.60 Table.xslt

表格處理的樣版因為所要處理的元素比較多，所以感覺起來比較複雜，事實上則不然。

處理 <table> 元素

(Table.xslt 第 46 到 59 行)

首先先說明如何設計處理 <table> 的 XSLT 範本。由圖 7.57 比較得知 <table> 是相對於 XSL-FO 表格元素中的 <fo:table-and-caption>，所以在範本中主要是由 <fo:table-and-caption> 構成。

在處理表格標題部份，由於表格標題是選擇性出現的元素，所以在處理範本中先行測試是否存在 <caption> 元素，如果存在則將 <caption> 轉換成爲 <fo:table-caption>，並將 <caption> 內容使用 <fo:block> 包裝起來。由於 <caption> 元素在範本中已經先行處理，所以處理 <caption> 的範本內容必須空無一物，以免呼叫 <xsl:apply-templates/> 時又再次的處理 <caption> 內容

(Table.xslt 第 60 行)。

處理 <col> 元素

(Table.xslt 第 61 到 63 行)

<col> 元素就是 XSL-FO 的 <fo:table-column> 元素，在處理上僅需要將替換成爲 XSL-FO 的 <fo:table-column> 元素即可，另外 <col> 元素中可能出現 width 屬性用來指定欄寬，對於 width 屬性需採用 <xsl:if> 來測試是否存在該屬性，如果存在該屬性則將之代入 <fo:table-column> 元素的 column-width 屬性之中。

處理 <thead>, <tfoot>, <tbody> 元素

(Table.xslt 第 64 到 78 行)

這三個元素分別是 XSL-FO 的 <fo:table-header>, <fo:table-footer>, <fo:table-body>，處理方式和處理 <col> 一樣，僅需要進行元素對元素替換的動作即可。

處理 <tr> 元素

(Table.xslt 第 79 到 83 行)

<tr> 元素的方式也和處理 <col> 的方式大同小異，只是 <tr> 元素替換成 XSL-FO 的 <fo:table-row> 元素即可。

處理 <th>, <td> 元素

(Table.xslt 第 84 到 119 行)

<th> 和 <td> 元素在 XSL-FO 的表格元素中均屬於 <fo:table-cell> 元素，所以處理上的差別僅在於儲存格呈現方式，例如 <th> 可以加上背景顏色，用來區隔和 <td> 內容的差別。由於 <th>, <td> 元素有時候會使用 colspan 屬性來合併相鄰的儲存格，爲了處理合併儲存格的 colspan 屬性，在樣版中特別呼叫了第 120 行的 cell-span 樣版，該樣版會將合併儲存格的屬性加入到 <fo:table-cell> 之中。

另外儲存格的內容有時候會依據 valign 屬性或是 align 屬性來調整內容在儲存格中顯式的位置，這兩個屬性若被指定時將分別會被替換成 display-align 與 text-align 兩種屬性。由於儲存格內可能還會有其他的樣式標記元素，所以使用 <xsl:apply-templates/> 繼續對內容標記的樣式進行處理，例如 或 <i> 等等。

下面是 Table.html 所產生的 XSL-FO 結果。


```

1 <?xml version="1.0" encoding="UTF-16"?>
2 <!DOCTYPE fo:root SYSTEM "FO.dtd">
3 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
4   <fo:layout-master-set>
5     <fo:simple-page-master master-name="A4" page-height="29.8cm"
6       page-width="21cm" margin="1.5cm 1.5cm 1.5cm 1.5cm">
7       <fo:region-body margin-top="1cm" margin-bottom="1cm"/>
8       <fo:region-after extent="1cm"/>
9     </fo:simple-page-master>
10  </fo:layout-master-set>
11  <fo:page-sequence master-reference="A4">
12    <fo:flow flow-name="xsl-region-body">
13      <fo:table-and-caption>
14        <fo:table-caption font-family="sans-serif"
15          text-align="center" space-before="3pt"
16          space-after="3pt" font-weight="bold"
17          keep-with-next.within-page="always">
18          <fo:block>The Caption Of Table</fo:block>
19        </fo:table-caption>
20        <fo:table table-layout="fixed"
21          space-before="12pt"
22          space-after="14pt">
23          <fo:table-header>
24            <fo:table-row>
25              <fo:table-cell>
26                <fo:block>Name</fo:block>
27              </fo:table-cell>
28              <fo:table-cell>
29                <fo:block>Age</fo:block>
30              </fo:table-cell>
31            </fo:table-row>
32          </fo:table-header>
33          <fo:table-body>
34            <fo:table-row>
35              <fo:table-cell>
36                <fo:block>Peter</fo:block>
37              </fo:table-cell>
38              <fo:table-cell>
39                <fo:block>19</fo:block>
40              </fo:table-cell>
41            </fo:table-row>
42            <fo:table-row>
43              <fo:table-cell>
44                <fo:block>Kevin</fo:block>
45              </fo:table-cell>
46              <fo:table-cell>
47                <fo:block>22</fo:block>
48              </fo:table-cell>
49            </fo:table-row>
50            <fo:table-row>
51              <fo:table-cell>
52                <fo:block>Wang</fo:block>
53              </fo:table-cell>
54              <fo:table-cell>
55                <fo:block>33</fo:block>
56              </fo:table-cell>
57            </fo:table-row>
58            <fo:table-row>
59              <fo:table-cell>
60                <fo:block>Helles</fo:block>
61              </fo:table-cell>
62              <fo:table-cell>
63                <fo:block>26</fo:block>
64              </fo:table-cell>
65            </fo:table-row>
66          </fo:table-body>
67        </fo:table>
68      </fo:table-and-caption>

```

```

69         </fo:flow>
70     </fo:page-sequence>
71 </fo:root>

```

圖 7.61 Table.html 所產生的 XSL-FO 結果

目前因為市面上的 XSL-FO 處理引擎對於 XSLFO 表格的處理非常薄弱，所以圖 7.61 的例子無法產生 PDF 輸出結果；但透過 XSL-FO 的 DTD 驗證結果為 Validate。

7.5.2.9 處理清單元素

在 HTML 中常用的兩種不同清單，分別是有序清單、和無序清單，這些種清單元素與 XSL-FO 的清單元素比較圖如下：

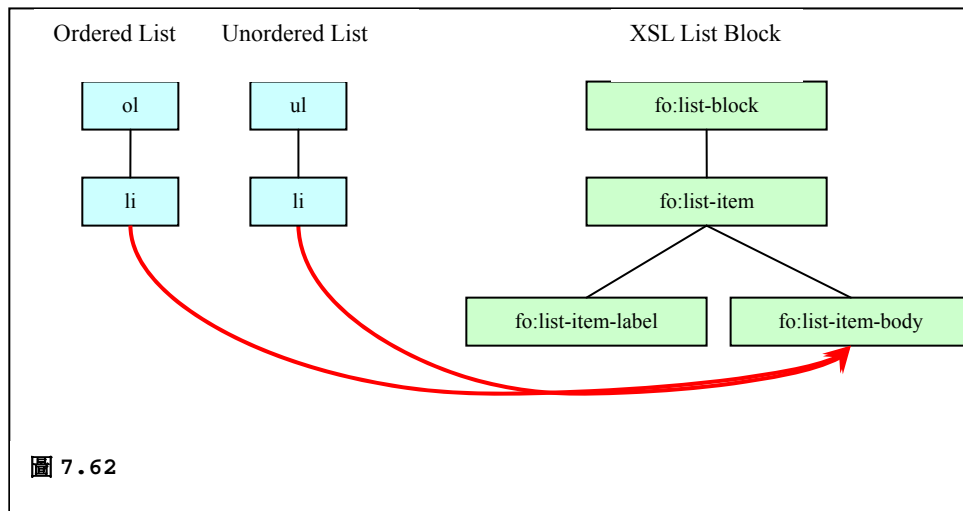


圖 7.62

很清楚的可以見到 `` 元素是等價於 XSL-FO 中的 `<fo:list-item-body>` 元素，所以設計處理樣版時，只需要將 `` 元素的內容填入 `<fo:list-item-body>`，而清單的標籤則依據有序清單填入序號或是無序清單填入符號至 `<fo:list-item-label>` 元素中即可。

7.5.2.9.1 Ordered List 的處理

下面是一個簡單的有序清單 HTML 範例：

```

1 <html>
2   <head>
3     <title>Processing Ordered List</title>
4   </head>
5   <body>
6     <ol>
7       <li>Violin</li>
8       <li>Viola</li>
9       <li>Cello</li>
10      <li>Double Bass</li>

```

```
11         </ol>
12     </body>
13 </html>
```

圖 7.63 OrderedList.html



圖 7.64 OrderedList.html 在瀏覽器中的結果

用來處理有序清單的 XSLT 樣式表如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3     xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4     xmlns:fo="http://www.w3.org/1999/XSL/Format">
5     <xsl:output method="xml" version="1.0" encoding="UTF-8"
6         indent="yes"/>
7     <xsl:template match="/">
8         <xsl:apply-templates/>
9     </xsl:template>
10    <xsl:template match="head"/>
11    <xsl:template match="body">
12        <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
13            <fo:layout-master-set>
14                <fo:simple-page-master master-name="A4"
15                    page-height="29.8cm"
16                    page-width="21cm" margin="1.5cm 1.5cm 1.5cm 1.5cm">
17                    <fo:region-body margin-top="1cm"
18                        margin-bottom="1cm"/>
19                    <fo:region-after extent="1cm"/>
20                </fo:simple-page-master>
21            </fo:layout-master-set>
22            <fo:page-sequence master-reference="A4">
23                <fo:flow flow-name="xsl-region-body">
24                    <xsl:apply-templates/>
25                </fo:flow>
26            </fo:page-sequence>
27        </fo:root>
28    </xsl:template>
29    <xsl:template match="ol">
30        <fo:list-block space-before="1cm"
31            provisional-distance-between-starts="1cm"
32            provisional-label-separation="0.4cm" start-indent="1cm">
33            <xsl:apply-templates/>
34        </fo:list-block>
35    </xsl:template>
36    <xsl:template match="ol/li">
37        <fo:list-item>
38            <fo:list-item-label end-indent="label-end()">
39                <fo:block>
40                    <xsl:choose>
41                        <xsl:when test="../@type = '>
42                            <xsl:number format="{../@type}'"/>
43                    </xsl:when>
```

```

40         <xsl:otherwise>
41             <xsl:number format="1."/>
42         </xsl:otherwise>
43     </xsl:choose>
44 </fo:block>
45 </fo:list-item-label>
46 <fo:list-item-body start-indent="body-start(">
47     <fo:block>
48         <xsl:apply-templates/>
49     </fo:block>
50 </fo:list-item-body>
51 </fo:list-item>
52 </xsl:template>
53 </xsl:stylesheet>

```

圖 7.65 OrderedList.xslt

圖 7.65 中的第 25 到 52 行是處理有旭青單元素的範本，由於有序清單有自動編號的功能所以在處理 `` 的範本中測試了 `` 的屬性 `type`，該屬性的內容依據 HTML4.01 的規格應該有下列五種類型。

Type	數字類型	範例
1	Arabic Numbers	1, 2, 3, ...
a	Lower Alpha	a, b, c, ...
A	Upper Alpha	A, B, C, ...
i	Lower Roman	i, ii, iii, ...
I	Upper Roman	I, II, III, ...

範本中將 `` 元素中的 `type` 屬性交由 `<xsl:number>` 元素處理，並將產生的結果包裝在 `<fo:list-item-label>` 中的 `<fo:block>`，這樣子就完成了有序清單的處理。

下面是 OrderedList.html 透過 OrderedList.xslt 產生的 XSL-FO 結果與 PDF 輸出畫面。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
3   <fo:layout-master-set>
4     <fo:simple-page-master master-name="A4"
5       page-height="29.8cm" page-width="21cm"
6       margin="1.5cm 1.5cm 1.5 1.5cm">
7       <fo:region-body margin-top="1cm" margin-bottom="1cm"/>
8       <fo:region-after extent="1cm"/>
9     </fo:simple-page-master>
10  </fo:layout-master-set>
11  <fo:page-sequence master-reference="A4">
12    <fo:flow flow-name="xsl-region-body">
13      <fo:list-block space-before="1cm"
14        provisional-distance-between-starts="1cm"
15        provisional-label-separation="0.4cm"
16        start-indent="1cm">
17        <fo:list-item>
18          <fo:list-item-label end-indent="label-end(">
19            <fo:block>1</fo:block>
20          </fo:list-item-label>
21          <fo:list-item-body start-indent="body-start(">
22            <fo:block>Violin</fo:block>
23          </fo:list-item-body>
24        </fo:list-item>
25      <fo:list-item>

```

```

26         <fo:block>2</fo:block>
27     </fo:list-item-label>
28     <fo:list-item-body start-indent="body-start()">
29         <fo:block>Viola</fo:block>
30     </fo:list-item-body>
31 </fo:list-item>
32 <fo:list-item>
33     <fo:list-item-label end-indent="label-end()">
34         <fo:block>3</fo:block>
35     </fo:list-item-label>
36     <fo:list-item-body start-indent="body-start()">
37         <fo:block>Cello</fo:block>
38     </fo:list-item-body>
39 </fo:list-item>
40 <fo:list-item>
41     <fo:list-item-label end-indent="label-end()">
42         <fo:block>4</fo:block>
43     </fo:list-item-label>
44     <fo:list-item-body start-indent="body-start()">
45         <fo:block>Double Bass</fo:block>
46     </fo:list-item-body>
47 </fo:list-item>
48 </fo:list-block>
49 </fo:flow>
50 </fo:page-sequence>
51 </fo:root>

```

圖 7.66 OrderedList.html 的 XSL-FO 輸出結果

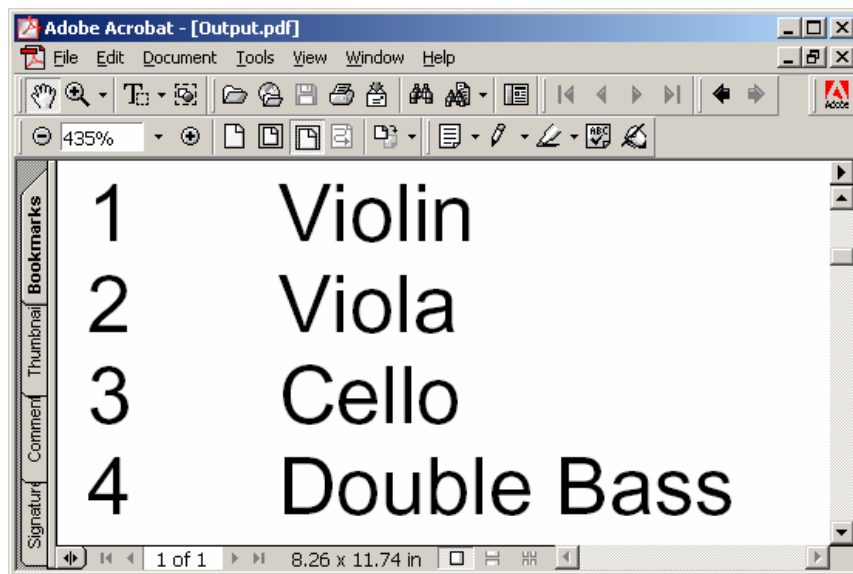


圖 7.67 OrderedList.html PDF 輸出結果

7.5.2.9.2 Unordered List 的處理

無序清單和有序清單的處理方式大同小異，差別只在於 `` 中的 `type` 屬性，根據 HTML 4.01 規格書，無序清單的 `type` 屬性有三種。

Type	樣式
disc	•

circle	○
square	■

所以只需針對 元素的處理範本做修改即可，修改後的範本如下：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0"
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4   xmlns:fo="http://www.w3.org/1999/XSL/Format">
5   <xsl:output method="xml" version="1.0" encoding="UTF-8"
6     indent="yes"/>
7   <xsl:template match="/">
8     <xsl:apply-templates/>
9   </xsl:template>
10  <xsl:template match="head"/>
11  <xsl:template match="body">
12    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
13      <fo:layout-master-set>
14        <fo:simple-page-master master-name="A4"
15          page-height="29.8cm"
16          page-width="21cm" margin="1.5cm 1.5cm 1.5 1.5cm">
17          <fo:region-body margin-top="1cm"
18            margin-bottom="1cm"/>
19          <fo:region-after extent="1cm"/>
20        </fo:simple-page-master>
21      </fo:layout-master-set>
22      <fo:page-sequence master-reference="A4">
23        <fo:flow flow-name="xsl-region-body">
24          <xsl:apply-templates/>
25        </fo:flow>
26      </fo:page-sequence>
27    </fo:root>
28  </xsl:template>
29  <xsl:template match="ul">
30    <fo:list-block space-before="1cm"
31      provisional-distance-between-starts="1cm"
32      provisional-label-separation="0.4cm" start-indent="1cm">
33      <xsl:apply-templates/>
34    </fo:list-block>
35  </xsl:template>
36  <xsl:template match="ul/li">
37    <fo:list-item>
38      <fo:list-item-label end-indent="label-end()">
39        <fo:block>
40          <xsl:choose>
41            <xsl:when test="../@type='disc'">
42              <xsl:text>·</xsl:text>
43            </xsl:when>
44            <xsl:when test="../@type='circle'">
45              <xsl:text>○</xsl:text>
46            </xsl:when>
47            <xsl:when test="../@type='square'">
48              <xsl:text>■</xsl:text>
49            </xsl:when>
50            <xsl:otherwise>
51              <xsl:text>·</xsl:text>
52            </xsl:otherwise>
53          </xsl:choose>
54        </fo:block>
55      </fo:list-item-label>
56      <fo:list-item-body start-indent="body-start()">
57        <fo:block>
58          <xsl:apply-templates/>
59        </fo:block>
60      </fo:list-item-body>

```

```

56         </fo:list-item>
57     </xsl:template>
58 </xsl:stylesheet>

```

圖 7.68 UnorderedList.xslt

7.5.3 XSLT 完整應用實例

本節將擷取 WC 網站上的 XSL 1.0 規格書的部份網頁內容，並應用本章所教導的 HTML 轉 XSL-FO 範本產生一份 Html2Fo.xslt，並把 XSL 1.0 規格書的部份網頁內容轉換成爲 PDF 檔案，此 Html2Fo.xslt 可以直接用使用在一般的 HTML 文件對 XSL-FO 轉換上。

W3C 的 XSL 1.0 規格書部份網頁畫面如下：

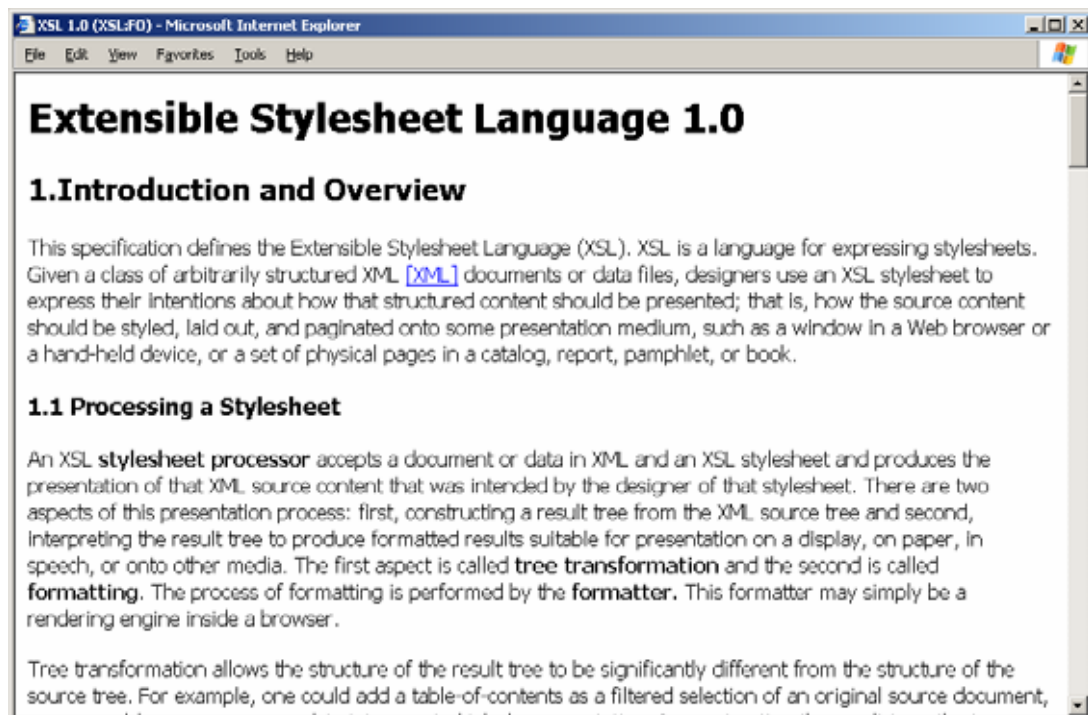


圖 7.69 XSL 1.0 規格書部份網頁內容

樣式表 Html2Fo.xslt 內容如下：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   xmlns:fo="http://www.w3.org/1999/XSL/Format">
4   <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
5   <xsl:template match="/">
6     <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
7       <fo:layout-master-set>
8         <fo:simple-page-master master-name="CoverPage"
9           page-height="15cm" page-width="20cm"

```

```

10         margin-top="0.5cm" margin-bottom="0.5cm"
11         margin-left="1cm" margin-right="1cm">
12         <fo:region-body margin-top="4cm"/>
13     </fo:simple-page-master>
14     <fo:simple-page-master master-name="LeftPage"
15         page-height="15cm" page-width="20cm"
16         margin-top="0.5cm" margin-bottom="0.5cm"
17         margin-left="0.5cm" margin-right="1cm">
18         <fo:region-before extent="1cm"/>
19         <fo:region-after extent="1cm"/>
20         <fo:region-body margin-top="1.1cm" margin-bottom="1.1cm"/>
21     </fo:simple-page-master>
22     <fo:simple-page-master master-name="RightPage"
23         page-height="15cm" page-width="20cm"
24         margin-top="0.5cm" margin-bottom="0.5cm"
25         margin-left="1cm" margin-right="0.5cm">
26         <fo:region-before extent="1cm"/>
27         <fo:region-after extent="1cm"/>
28         <fo:region-body margin-top="1.1cm" margin-bottom="1.1cm"/>
29     </fo:simple-page-master>
30     <fo:page-sequence-master master-name="Contents">
31         <fo:repeatable-page-master-alternatives>
32             <fo:conditional-page-master-reference
33                 master-reference="LeftPage" odd-or-even="even"/>
34             <fo:conditional-page-master-reference
35                 master-reference="RightPage" odd-or-even="odd"/>
36         </fo:repeatable-page-master-alternatives>
37     </fo:page-sequence-master>
38 </fo:layout-master-set>
39 <fo:page-sequence master-reference="CoverPage">
40     <fo:flow flow-name="xsl-region-body">
41         <fo:block font="Arial" font-size="32pt"
42             text-align="center" space-after="14pt">
43             Extensible Stylesheet Language Version 1.0</fo:block>
44         <fo:block text-align="center" space-before="50pt">
45             <fo:external-graphic src="file:w3c_home.bmp"/>
46         </fo:block>
47         <fo:block space-before="1cm" font-family="Times New Roman, Times New
Roman"
48             font-size="13pt" text-align="center">W3C Recommendation 15 October
2001
49         </fo:block>
50     </fo:flow>
51 </fo:page-sequence>
52 <fo:page-sequence master-reference="Contents" initial-page-number="2">
53     <fo:static-content flow-name="xsl-region-before">
54         <fo:block font-size="10pt" text-align="center">
55             Extensible Stylesheet Language Version 1.0</fo:block>
56     </fo:static-content>
57     <fo:static-content flow-name="xsl-region-after">
58         <fo:block font-size="10pt" text-align="center">Page #<fo:page-number/>
59         </fo:block>
60     </fo:static-content>
61     <fo:flow flow-name="xsl-region-body">
62         <xsl:apply-templates/>
63     </fo:flow>
64 </fo:page-sequence>
65 </fo:root>
66 </xsl:template>
67 <xsl:template match="title">
68     <xsl:apply-templates/>
69 </xsl:template>
70 <xsl:template match="body">
71     <xsl:apply-templates/>
72 </xsl:template>
73 <xsl:template match="h1">
74     <fo:block font-size="16pt" font="Times New Roman" font-weight="bold"
75         color="blue" space-before="10pt" space-after="10pt">
76         <xsl:value-of select="."/>
77     </fo:block>

```



```

78     </xsl:template>
79     <xsl:template match="h2">
80         <fo:block font-size="14pt" font="Times New Roman" font-weight="bold"
81             space-before="14pt" space-after="10pt">
82             <xsl:value-of select="."/>
83         </fo:block>
84     </xsl:template>
85     <xsl:template match="h3">
86         <fo:block font-size="12pt" font="Times New Roman" font-weight="bold"
87             space-before="14pt" space-after="10pt">
88             <xsl:value-of select="."/>
89         </fo:block>
90     </xsl:template>
91     <xsl:template match="h4">
92         <fo:block font-size="12pt" font="Times New Roman" space-before="14pt"
93             space-after="10pt" font-weight="bold">
94             <xsl:value-of select="."/>
95         </fo:block>
96     </xsl:template>
97     <xsl:attribute-set name="Paragraph">
98         <xsl:attribute name="text-indent">1.2cm</xsl:attribute>
99         <xsl:attribute name="font-family">Times New Roman</xsl:attribute>
100        <xsl:attribute name="font-size">10pt</xsl:attribute>
101    </xsl:attribute-set>
102    <xsl:template match="p">
103        <fo:block xsl:use-attribute-sets="Paragraph">
104            <xsl:apply-templates/>
105        </fo:block>
106    </xsl:template>
107    <xsl:template match="b">
108        <fo:inline font-weight="bold">
109            <xsl:apply-templates/>
110        </fo:inline>
111    </xsl:template>
112    <xsl:template match="i">
113        <fo:inline font-style="italic">
114            <xsl:apply-templates/>
115        </fo:inline>
116    </xsl:template>
117    <xsl:template match="u">
118        <fo:inline text-decoration="underline">
119            <xsl:apply-templates/>
120        </fo:inline>
121    </xsl:template>
122    <xsl:template match="a">
123        <xsl:variable name="AnchorText">
124            <xsl:value-of select="."/>
125        </xsl:variable>
126        <xsl:apply-templates/>
127        <xsl:if test="@href!=\$AnchorText">
128            <fo:inline>
129                <xsl:text></xsl:text>
130                <xsl:value-of select="@href"/>
131                <xsl:text></xsl:text>
132            </fo:inline>
133        </xsl:if>
134    </xsl:template>
135    <xsl:template match="ol">
136        <fo:list-block space-before="1cm"
137            provisional-distance-between-starts="1cm"
138            provisional-label-separation="0.4cm" start-indent="1cm">
139            <xsl:apply-templates/>
140        </fo:list-block>
141    </xsl:template>
142    <xsl:template match="ol/li">
143        <fo:list-item>
144            <fo:list-item-label end-indent="label-end()">
145                <fo:block>
146                    <xsl:choose>

```

```

147         <xsl:when test="../@type">
148             <xsl:number format="{../@type}"/>
149         </xsl:when>
150         <xsl:otherwise>
151             <xsl:number format="1."/>
152         </xsl:otherwise>
153     </xsl:choose>
154 </fo:block>
155 </fo:list-item-label>
156 <fo:list-item-body start-indent="body-start()">
157     <fo:block>
158         <xsl:apply-templates/>
159     </fo:block>
160 </fo:list-item-body>
161 </fo:list-item>
162 </xsl:template>
163 <xsl:template match="ul">
164     <fo:list-block space-before="1cm"
165         provisional-distance-between-starts="1cm"
166         provisional-label-separation="0.4cm" start-indent="1cm">
167         <xsl:apply-templates/>
168     </fo:list-block>
169 </xsl:template>
170 <xsl:template match="ul/li">
171     <fo:list-item>
172         <fo:list-item-label end-indent="label-end()">
173             <fo:block>
174                 <xsl:choose>
175                     <xsl:when test="../@type='disc'">
176                         <xsl:text>|'</xsl:text>
177                     </xsl:when>
178                     <xsl:when test="../@type='circle'">
179                         <xsl:text>|³</xsl:text>
180                     </xsl:when>
181                     <xsl:when test="../@type='square'">
182                         <xsl:text>|½</xsl:text>
183                     </xsl:when>
184                     <xsl:otherwise>
185                         <xsl:text>|'</xsl:text>
186                     </xsl:otherwise>
187                 </xsl:choose>
188             </fo:block>
189         </fo:list-item-label>
190         <fo:list-item-body start-indent="body-start()">
191             <fo:block>
192                 <xsl:apply-templates/>
193             </fo:block>
194         </fo:list-item-body>
195     </fo:list-item>
196 </xsl:template>
197 </xsl:stylesheet>

```

圖 7.70 Html2Fo.xslt

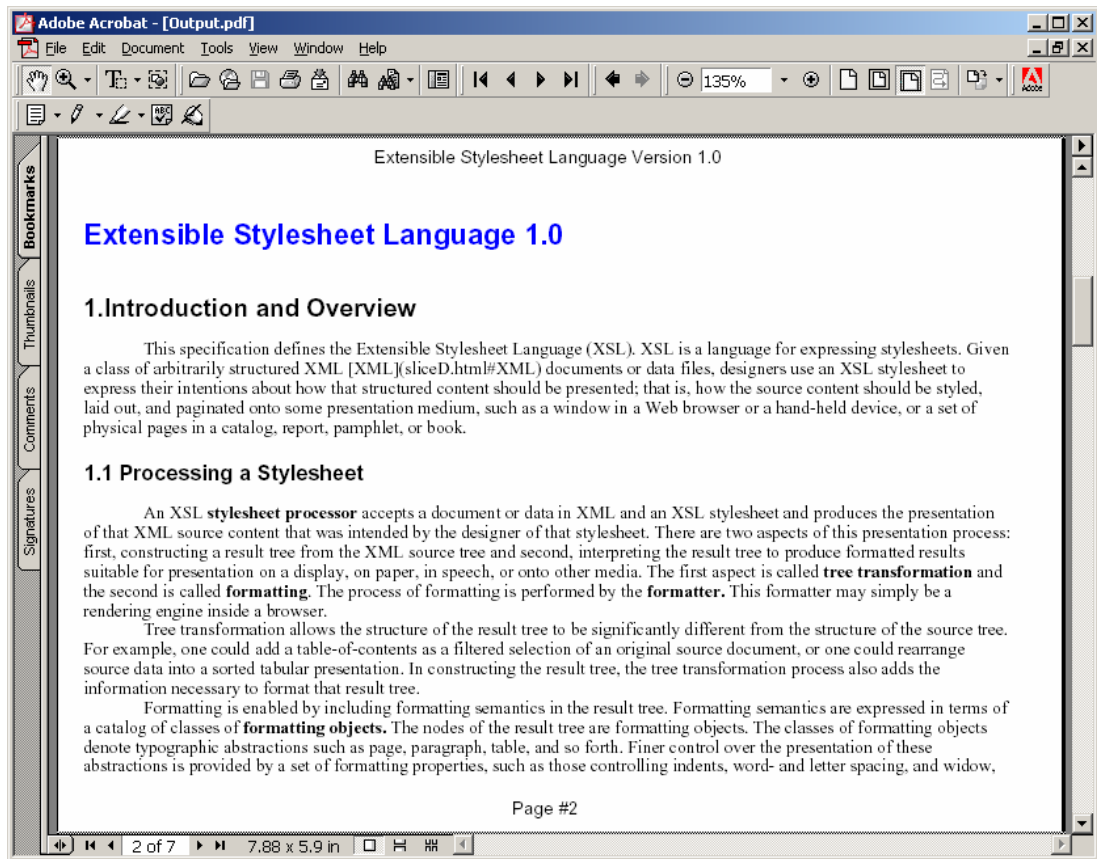


圖 7.71 PDF 輸出結果