

第 2 章相關技術探討

本章主要是相關文獻的整理與探討。第一節介紹資訊隱藏的特性，和加密有何區別。第二節介紹浮水印與圖形格式，並說明傳統浮水印如何隱藏資訊到各種點陣圖。第三節介紹 XML 文件的應用技術。第四節介紹 DOM Tree 的產生及應用。第五節介紹可變動向量圖形（Scalable Vector Graphics）的特性，並敘述實作因素考量的層面，以及常用語法。

2.1 資訊隱藏

古時候要把極機密的訊息從甲方送到乙方，可能藉由人力、獸力、大自然的力量，傳遞的方法隨著時間而日趨複雜，尤其是在戰爭期間，發展了不少資訊隱藏的方法。隨著多媒體與電腦網路技術的普及，資訊隱藏的技術有日新月異的發展，許許多多資訊隱藏在浩瀚網海中，人們可以輕易的藉由網路傳送訊息，包括文字、語音、靜態圖片與動態影像等，不但訊息的傳播快速，也刺激了新的技術的產生。

有關資訊隱藏，在學術界上較為常用的術語有 Data Hiding、Information Hiding、Steganography [9][12][14]等三個，其中前兩個名詞我們較易從字面上獲得其中的涵意，就是將資料或者是資訊隱藏起來的意思，兩者如果真要細分，可從 Data 與 Information 這兩個字去分辨，但在學術界上，對於這些名詞間的分別，並沒有多去注意[8]。

至於 Steganography 這個名詞，則可回溯到西元 1499 年，由 Trithemius 將 steganos 與 graphein 兩個希臘字合併而成，steganos 的字義是 covered，是遮

蔽、隱藏的涵意，graphein 則是 to write，兩個字合併起來，就是將所寫的字隱藏起來的意思。Steganography 這個字後來因為用法多和魔術、妖術 (Black art) 產生關連，所以漸漸較少被用到。直到近來在 David Kahn 所著的 Codebreakers 一書中被賦予新的意義後才又再度起用。

在古希臘時代的人，有用過一種方法隱藏訊息，就是把侵略的警訊塗寫在書寫板蠟質層下方的木頭上——於是書寫板本身看起來就像是空白的。只有目標收訊者知道，而這張看起來空白的書寫板蠟質層下方卻帶有訊息。古時候隱藏訊息的另一種方法，是把信差的頭剃光，而機密訊息就刺青在他的頭皮上。然後派遣信差出去，在前往目的地途中，他的頭髮就會長回來，抵達之後再把頭髮剃掉，訊息就會顯露出來。

第二次世界大戰期間，資訊隱藏最常見的形式，就是使用隱形墨水寫下機密訊息。當時用的「隱形墨水」常常是牛奶、果汁或尿液。把訊息一加熱，隱形墨水和寫在上面的機密訊息就會顯露出來。

隨著科技的發展，傳送機密訊息的方法也越來越先進。今日的恐怖分子再也不必擔心留下活動的證據，因為訊息現在可以藏在 Internet 上了。這和透過電子郵件傳送的加密訊息不同，資訊隱藏不會留下傳送者或收件者的記錄。

今日資訊隱藏的高科技形式，包括把機密訊息隱藏在 Internet 上的影像或聲音檔案之中。如果要在一個電腦檔案（圖形、音訊、文字、HTML 或甚至軟碟磁片）內插入訊息，就必須使用專門的資訊隱藏軟體。這種軟體很容易取得——很多下載網站上都有這類免費軟體或共享軟體。軟體會把祕密的資訊弄亂，混進影像或聲音檔案中已存在但不重要的位元內，而這種變化不夠明顯，人類的眼睛或耳朵偵測不到。訊息可能只含有一位元偵測不到的聲音，或是在一張 JPEG 影像中

改變幾個像素。在漫不經心的觀察者看來，影像或聲音的位元組看起來或聽起來並無任何異樣。他們幾乎完全不知道，網頁上其實埋有隱藏訊息。這些影像只有利用收信者選定的「私密金鑰」或密碼才能解開。否則，這些訊息是不可能看到、讀到或聽到的。

資訊隱藏與密碼學有何差異

至於密碼學 (Cryptography) 這個字也是由兩個希臘字根所組成，Crypto 這個希臘字義是 secret 的意思，因此 Steganography 的意義相當接近於 Cryptography，也有學者將其視為是 Cryptography 的分支。我們可以用一個簡單的方法來區別這兩個字，Cryptography 是將訊息 (Message) 的意義 (Meaning) 隱藏起來，讓你看不懂其涵意；至於 Steganography 則是將訊息本身的存在性 (Existence) 隱藏起來，讓你根本不曉得有訊息的存在。

密碼學可改變檔案或訊息的內容，讓除了目標收件者以外的每個人都無法閱讀。目標收件者握有「金鑰」，可以解開加密檔案的鎖，並以傳送者計劃的方式來檢視檔案。加密的訊息並不是隱藏的，而且訊息送進送出都有可能受到偵測或監視。一旦找出加密的方法，仍然需要破解密碼的人來找出金鑰，才可能將訊息解密。

可以把資訊隱藏想成一種非常堅固的加密方式。而它隱藏訊息的方式，會讓觀察者也許根本察覺不到有訊息的交換。它和密碼學不一樣，資訊隱藏是不易偵測到的。人們常常利用資訊隱藏來輔助加密。透過加密，再結合加密資料隱形，就能徹底保護訊息而不讓資料間諜得手。

許多企業和政府或軍方的內部檔案已經禁止隨意複製或限制使用而加密，這更造成資訊隱藏技術在 Internet 上蓬勃發展。資訊隱藏軟體的取得與使用都相當容易，而且每天都有幾百萬個新的音訊和影像檔案貼在 Internet 上。那麼多不同

類型的檔案，都可能含有各式各樣的隱藏資訊，而追蹤或找到這些檔案，幾乎是不可能的任務。也許我們已經看過某個含有隱藏訊息的網頁、圖像，但我們卻從來不會去注意、發現並解開資訊隱藏，揭發這些秘密簡直就像在全球資訊網的浩瀚大海中撈針。

2.2 浮水印與圖形

數位浮水印技術 (digital watermarking) [2] 是透過改變數位內容本身一部份或整體資料而放入其他資訊的技術，當然數位內容的品質不能被降格太多，以保持其價值。

最常被使用的浮水印是可見的浮水印 (Visible Watermark)，如下圖 2-1 所示，雖然這是最顯而易見的版權宣示方法，缺點是會破壞數位內容的品質，而且只能用人眼來偵測，無法用電腦程式或裝置做自動化的偵測。



圖 2 - 1 Visible Watermark

當我們加入浮水印的同時，多少會改變到原來圖檔的情況（比如加入一個商標 trademark 或一串文字），那就破壞了被保護圖檔的價值與可觀度。但是，為了

能防止浮水印被輕易移除的可能性，故在製作圖像浮水印時，必須在色彩品質不失真與不可被移除的情形之間取得一個平衡；也因此，浮水印的不可見(invisible)性就顯得相當重要。如此一來，隱藏式數位浮水印的重要性與必然性就產生了。

不可見的浮水印則必須兼顧所要藏的資料和原始檔案的品質，目前有數種方法，如：Pixels Luminance、DCT Coefficient、Wavelet Coefficient、展頻通訊(Spread-Spectrum Communication)、和雜訊理論(Noisy Theory)等。依據應用的不同，又可分為公開型浮水印(Public Watermark)、破壞型浮水印(Fragile Watermark)和隱私型浮水印(Private Watermark/Secure Watermark)。

在技術上，數位浮水印可被表示成一種雜訊樣式，然後將它加入原始的資訊中。這些被表示成“雜訊”的浮水印或其被加入的位置，可以由所使用方法或所使用的金鑰來決定。我們以下圖 2-2 用雜訊來表示浮水印的觀念，其步驟分別是：
(1)將浮水印轉換成“雜訊”形式(2)將此雜訊形式之浮水印加入原來的影像中(3)加入浮水印後的影像基本上與原來影像沒有差異，且浮水印為肉眼觀查不到的，並僅能用抽取軟體將之還原。

Watermarks: Secret Code for Protection

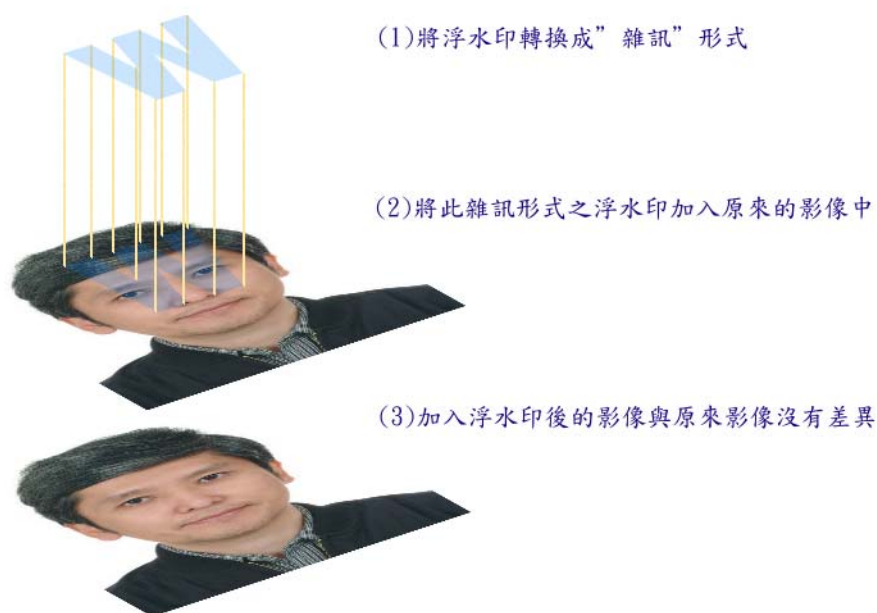


圖 2 - 2 Invisible Watermark

公開型浮水印的主要功能不在保護內容，而是提供版權資料和版權清算之用，缺點是無法藏太多的資料，也很容易被破解。破壞型浮水印可以讓使用者檢查內容的完整性，辨識壓縮檔和遭竄改檔案之間的不同。隱私型浮水印是以私鑰 (Private Key) 來保護內容，當盜用者想除掉浮水印時，必定會破壞或降低內容的品質，當然如果他試圖用 Low Pass Filter[7]、Cropping[11] 或 Noise Addition[6] 等攻擊方法，一來品質會降低，二來他並不知道浮水印是否已經真正被除掉，因為他沒有私鑰。但是科技是日新月異，世界上沒有破不了的浮水印，只要浮水印的設計者能讓盜用者付出高代價和風險，他便成功了。除了技術的問題之外，目前各國在法庭上還很難說服法官接受浮水印做為證據，也是浮水印一直無法廣泛被使用的原因之一。

2.3 簡述 XML

XML(Extensible Markup Language)，中文稱之為『可延伸的標記語言』，是國際標準組織(Internation Standard Organization, ISO)公布的標準通用標示語言(Standard Generalized Markup Language, SGML)的子集合。XML 的出現，讓網路應用向前邁進一大步，使資訊儲存方法能超過時間和空間的限制，跨過不同電腦平台，達成資訊流通與共享。

資料庫的發展，從 1990 年到 2000 年的 10 年之間是 server-client 全盛時期，各式各樣的大小電腦平台，擁有自己獨立的資料儲存方式，從大到數百萬美元的 mainframe，小到掌中型 PDA，系統之間很難傳遞資料，但是透過 XML 當作交流媒介，可順利跨過系統差異的門檻。2001 年起資料庫的明日之星，已是 XML 格式的文件。

資料庫可以不再過度依賴集中式的 SERVER，可以用 XML 文件格式分散在網際網路上，傳遞 XML 文件只要裝上 XML 解析器(Parser)，便可解讀由別台電腦所傳來的 XML 資訊，很多額外資訊的處理便可在本地端完成，減輕資料庫伺服器的負擔，抒解了伺服器對外交通流量，因而提高服務效能。不同平台之間不必擔心看不懂對方的特定資料格式，即使日後有一方改變平台或應用程式，仍然可以傳遞 XML 格式的資料，也不會影響和另一方平台的交流。

XML Parser 專門對 XML 文件作解析，是介於 XML 檔案和應用程式之間的低階軟體工具。如果 XML 文件有結構上的錯誤，或語法的錯誤，Parser 會傳回錯誤訊息到 Browser 上，程式開發者可以針對錯誤去逐條修改文件，直到沒有錯誤為止。解析後的 XML 文件會產生一個樹狀邏輯結構在記憶體中，和原有的 XML

文件完全對應，應用程式可能是 Browser、商用程式、Database，都根據這個樹狀邏輯結構，來處理或應用其中的資料。

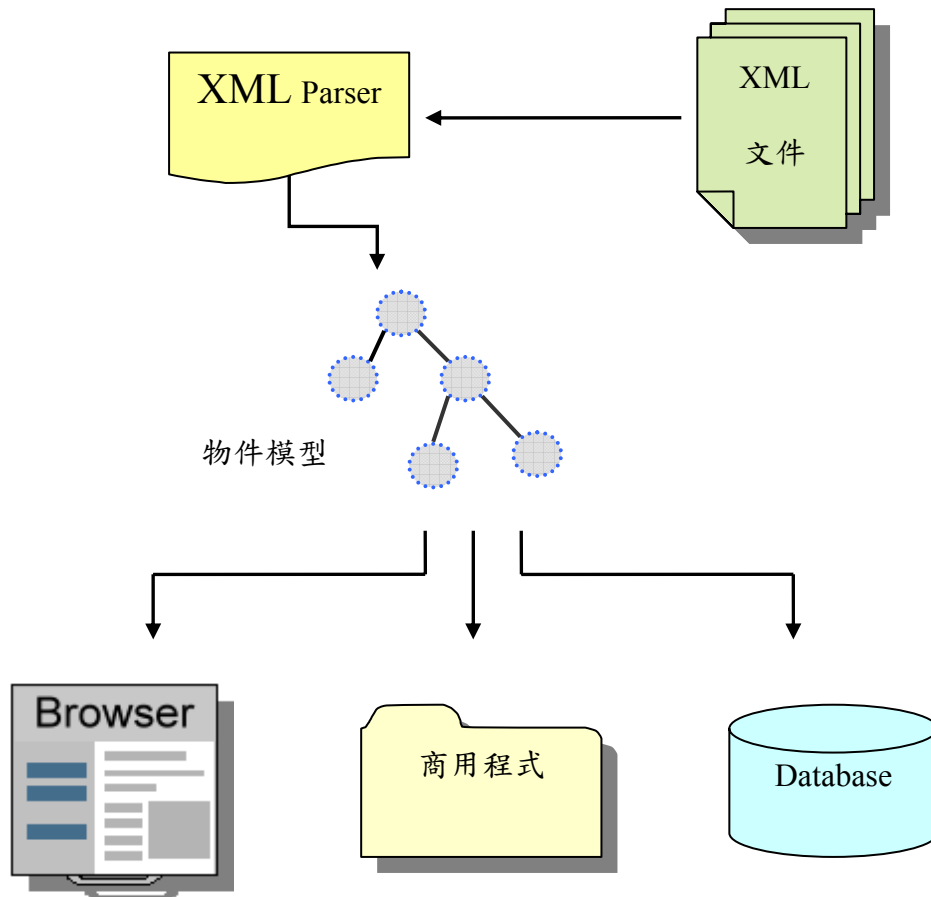


圖 2-3 XML 文件處理步驟

電腦普及至今，很多叱吒一時的主流、先進文件軟體，都經不起時間嚴格的考驗。客戶端要求越趨嚴格，更進步的文件軟體搭配更新的硬體，提供更強大的功能，但舊的格式文件形成包袱，經過多年後，老舊的機器、軟體不易維護文件，又難以即時更新。XML 文件卻沒有這樣的問題，因為他的儲存格式是純文字，對於文件的保存及永續應用佔有絕對的優勢，想要將 XML 格式的文件搬移、整理到其他系統是輕而易舉，轉成其他格式的文件也唾手可得。目前社會上有很多

文件要長久保留下去，舉凡各種法律條文、醫療病例、政府公文、歷史檔案、學術報告、乃至於個人日記、回憶錄等，只要是需要長期保存的資訊，都可藉由 XML 的特性，延續資訊的壽命至未來。

使用一般簡單的文書編輯器，就可以編輯 XML 檔案，它和網頁常用的 HTML 文件有些類似，但不同的事 XML 有較嚴謹的定義[1]：

1. XML 文件中，必須先作版本宣告，最好還能包含編碼(encoding)的宣告。
2. XML 文件中，只能有一個根元素，其他的元素都包含在此根元素之下。
3. XML 文件中，每一個起始標籤(Begin Tag)都必須有一個對應的結尾標籤(End Tag)。
4. 不含內容的空標籤，必須以<標籤名/>表示，也就是說在標籤名稱後面加上"/"。
5. 屬性必須以雙引號括起來。
6. 標籤名稱大小寫視為不同。
7. 標籤之間不得交叉。

XML 程式碼範例如下，由此程式碼範例可清楚見到由使用者自創的標籤，當有需要時就可以依需求自行定義一個新的標籤，此即是『可延伸的標籤語言』。

```
<?xml version="1.0" encoding="big5"?>
<member>
  <張大智>
    <sex>boy</sex>
    <email>large@ice.ntnu.edu.tw</email>
    <address>高雄市尚義街</address>
    <password>123456</password>
  </張大智>
  <柯小薇>
    <sex>girl</sex>
    <email>wei@mail.ice.ntnu.edu.tw</email>
    <address>台北縣中和市中正路</address>
    <password>987654</password>
  </柯小薇>
  <james>
    <sex>boy</sex>
    <email>james@hcshtp.edu.tw</email>
    <address>1132S. 3rd St. Alhambra, CA 91803</address>
    <password>qwerty</password>
  </james>
</member>
```

在這個 XML 程式中，描述三個會員的資料，並且記錄了每一個人的 sex、email、address、passwd，如下圖 2-4：

member	
▲ 張大智	
sex	boy
email	large@ice.ntnu.edu.tw
address	高雄市尚義街
password	123456
▲ 柯小薇	
sex	girl
email	wei@mail.ice.ntnu.edu.tw
address	台北縣中和市中正路
password	987654
▲ james	
sex	boy
email	james@hcsh.tp.edu.tw
address	1132S. 3rd St. Alhambra, CA 91803
password	qwerty

圖 2 - 4 XML 文件紀錄 member 會員資料

在沒有特別安排的版面之下，瀏覽器端直接呈現的畫面如下圖 2-5：

```

<?xml version="1.0" encoding="big5" ?>
- <member>
- <張大智>
  <sex>boy</sex>
  <email>large@ice.ntnu.edu.tw</email>
  <address>高雄市尚義街</address>
  <password>123456</password>
</張大智>
- <柯小薇>
  <sex>girl</sex>
  <email>wei@mail.ice.ntnu.edu.tw</email>
  <address>台北縣中和市中正路</address>
  <password>987654</password>
</柯小薇>
- <james>
  <sex>boy</sex>
  <email>james@hcsh.tp.edu.tw</email>
  <address>1132S. 3rd St. Alhambra, CA
  91803</address>
  <password>qwerty</password>
</james>
</member>

```

圖 2 - 5 XML 文件直接呈現的畫面

XML 的特色

XML 的結構化的資訊內容導向，可供其他電子資料傳遞、文件出版、電腦輔助設計或製造、資料庫管理等系統，在處理重複資料和共享的資料時，能有效提升效能，這種方法能將資訊內容、結構和格式都不同的文件要素加以區分，且同時能保存文件的資料和結構，卻又不指出文件的呈現格式，因此 XML 技術上的優勢和特色，使資訊交換產生了重大的改變，並在應用上提供更多的可能性，因此歸納 XML 的特色如下：

1. 由 SGML(Standard Generalized Markup Language)[15]發展而來的簡單標示語言
2. 一種 Meta Language,而非單一標籤的標示語言
3. 對於同一份內容只需傳輸一次,便可重複使用
4. 更強的資料連結功能,可做雙向甚至多向資料連結
5. 可自定文件類別,處理 HTML 無法設計的專門格式文件
6. 支援 CSS,XSL,DOM

XML 的安全機制

隨著 XML 的風行，IETF(The Internet Engineering Task Force) 於是在 2001 年三月將 XML 數位簽章納入 Internet 的標準(RFC 3075)。XML Encryption 是由 W3C 所推動 XML 加密規格，它分別由下列處理方式所組成：

1. XML 加密需求 (XML Encryption Requirements): 列示了 XML 加密標準的設計原則、範疇及需求等，其需求包含了相關的加密語法、資料模組、格式、加密處理及外觀需求等。
2. XML 加密語法及處理程序 (XML Encryption Syntax and Processing): 定義了如何對資料進行加密處理並以 XML 呈現結果。加密的資料可能是任

意形式的 XML 格式資料，加密後的資料為稱為 EncryptedData 密文資料，EncryptedData 密文會取代加密的 XML 元素及內容資料的原資料。

3. XML 電子簽章解密轉換 (Decryption Transform for XML Signature): 具體說明了如何使用 XML 簽章 (XML Signature) [14] 來解密轉換，使用 XML 簽章來識別 XML 加密的結構，區別加密後簽章或加密前簽章以作為簽章的驗證。

XML Encryption 制訂了 XML 文件的加密的規範，透過數位的加密能夠安全的保存及傳送機密的數位資訊，其加密方式可以針對 XML 文件的部分區段進行加密，和先前的加密方式不同，例如說：在一份交易表單裡有信用卡卡號，我們可針對該卡號進行加密即可，不需要對整份文件進行加密，而提高使用的效率。

現有 W3C 對於 XML 應用於電子商務的安全機制，只有單純的資料加密功能，這樣的安全機制無法適用於全部的電子商務環境。因為透過密文方式傳送資料，雖然可以避免發送方所傳送的資料內容外洩，但無法阻止攻擊者對於資料攔截破壞，而讓接收方無法收到資料。除此之外對於圖像的智慧財產權，雖然已有 Watermark，但大都針對於點陣圖，對於向量圖浮水印的研究比較稀少，對於 XML 為基礎的 SVG 的格式目前更是沒有相關的研究，W3C 也沒有提出有效的解決方案，也因此為上述問題，我們應用了資訊隱藏於 XML 的應用，也提出了一個如何保護 SVG 智慧財產權的方式。

2.4 簡述 DOM

DOM(Document Object Model)的相關標準是由 W3C(World Wide Web Consortium)所制訂，是一種跨平台的 XML 應用程式介面，允許程式設計者利用 Script 語言進行動態存取文件，可即時更新 HTML 與 XML 文件的內容和架構。DOM 被分成幾個 Level，而非用版本編號來表示其發展進程。DOM Level 1 在 1998 年底就已經是 W3C 推薦使用的標準，使用 DOM 來表示的文件不限於 XML、HTML 或其他內容模型都可以。

DOM Level2 在 2002 年 11 月成為 W3C 推薦使用的標準，針對特定內容模型，諸如 XML、HTML、CSS、SVG 提供一些模組和選項，特別是加入名稱空間(Name Space)的支援，容許設計者使用來自不同來源或帶有不同目的的資訊，而不會發生衝突。同時以 Level1 為基礎而附加其上面，也提供標準的 API 可以新增、刪除、查詢與修改 DOM Tree 的內容，將所得結果輸出成 XML 文件。

W3C 在 2004 年陸續發表 DOM Level3，將 Core、Load and Save、Validation 等部分，對外公告為 W3C 的 Recommendation。DOM Level3 的 Load and Save 算是標準 API，但目前 XML Parser 對 DOM Level3 的支援程度還不算太好。

DOM 的結構

一份 XML 文件經由 DOM 處理過後呈現的樣式，經由底下範例可清楚解釋：

```
<?xml version="1.0" encoding="big5"?>
<books>
  <決斷 2 秒間>
    <author>葛拉威爾</author>
    <publisher>時報出版</publisher>
    <price>300</price>
```

</決斷 2 秒間>

<禮物>

<author>莊靜君</author>

<publisher>平安文化</publisher>

<price>199</price>

</禮物>

<死亡印記>

<author>杜蕾蕾</author>

<publisher>宏道文化</publisher>

<price>320</price>

</死亡印記>

</books>

此份 XML 文件的樹狀結構，如圖 2-6 如示：

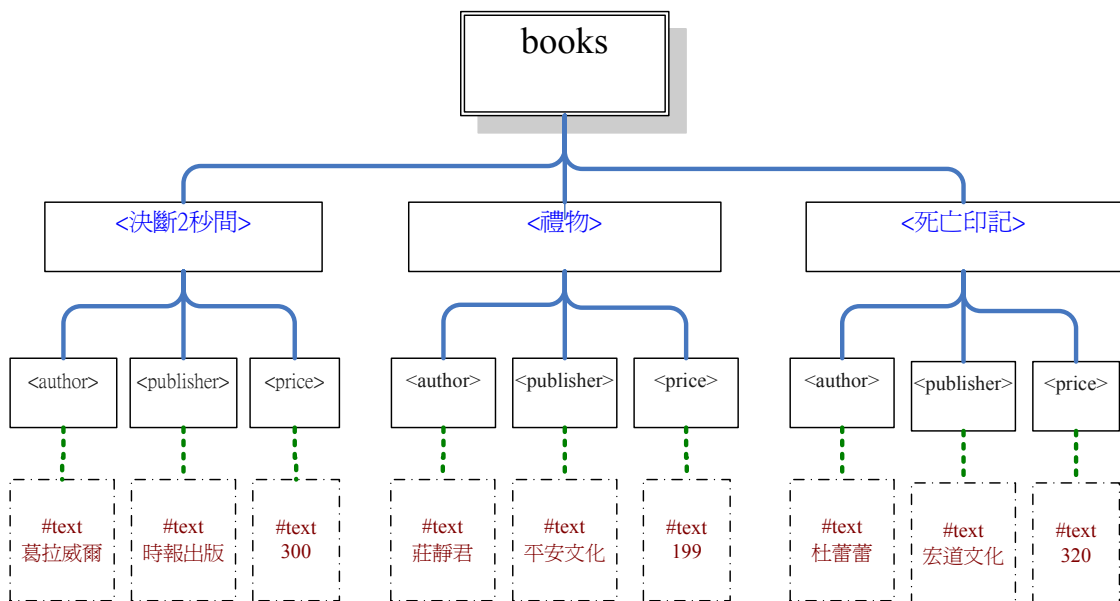


圖 2 - 6 XML 文件的樹狀結構圖

XML 最常見的節點類型分成：Document、Element、Attribute、Text；而不常見的節點類型還有：CDATA、註釋、處理指令和文件片段。DOM 是一些節點所形成的集合，將一些常用節點組合成下列七種 DOM 的常用物件：

DOM 的常用物件

1. Node 物件

在 DOM 的樹狀結構中，最重要的事 Node 物件(節點物件)，一份 XML 文件經由 DOM 處理後，根據 XML 標籤內的性質將整個結構化成樹狀，有些類型節點可以擁有子節點，有些就只能當葉節點，其後面不能再有 其他節點。

2. NodeList 物件

這是一群 Node 物件的集合，是整個樹狀結構下的一段子節點樹。

3. NameNodeMap 物件

NameNodeMap 物件和 NodeList 物件都是代表節點的集合，與 NodeList 不同之處是 NameNodeMap 通常用來代表一群 Attribute 物件的集合。

4. Document 物件

一份 XML 文件最上層的節點，亦即根節點。

5. Element 物件

指的是 XML 文件中的元素節點，如果元素含有文字，則對應的就是一個 Text 物件；如果元素含有屬性，則對應的就是一個 Attribute 物件。

6. Attribute 物件

Attribute 物件代表的是 Element 物件的屬性。

7. Text 物件

Text 物件代表的是文字節點或是屬性節點所對應的文字內容。

2.5 可變動向量圖形 (SVG)

2.5.1 SVG 簡介

SVG (Scalable Vector Graphics), 1.0 版由 W3C 於 2001 年 9 月正式正式推薦，是一種以 XML 語言為基礎的新圖檔標準，主要用來改善網頁圖片的品質，可以讓瀏覽器上的圖檔更有效率、更多效果。SVG 可讓網頁上顯示出各種高品質的向量圖形，同時也支援文字、動畫、遮罩效果，對幾何圖形色彩、透明度、變形、修剪路徑、濾鏡效果都有充分支援。

網站的設計通常以點陣圖來處理圖形，若要達到檔案小、傳遞快，最好還有動畫，則考慮使用向量圖形，新興於網路的 2D 向量圖格式有兩種，其一為 SVG 圖形格式，從瀏覽器觀賞網站的 SVG 圖形，需透過 Adobe 公司的外掛軟體 SVG Viewer；另一個是 swf 圖形格式，也需要在瀏覽器安裝 Macromedia 公司的 Flash player。這兩種向量圖形各有優缺點，前者是開放標準的純文字向量式向量圖形描述語言，後者佔有廣大市場，但為封閉格式。

2.5.2 SVG 的特色

由於 SVG 是 XML 的子集合，因此具備了一般網頁傳統使用圖形如 JPEG、PNG、GIF、BMP 等所沒有的特性：檔案小、傳輸速度快，還可支援下列各項：Type 1 及 TrueType 字型、XML (延伸標記語言)、CSS (樣式表)、互動式動作，以及動態 HTML 動畫。

對 SVG 而言，最大的優點當然是和其它開放標準相容，因此 SVG 擁有許多的特色分別介紹如下：

1. 基於 XML 標準

微軟的 office 套裝軟體無疑是最成功的商用軟體，而大量支援 XML，是微軟

標榜 Office 2003 的賣點之一，軟體巨人的行動已經承認 XML 是下一代網路標記語言，與 HTML 一樣，XML 也源 SGML，它擁有 HTML 語言所缺乏的伸縮性與靈活性。SVG 設計之初就是以 XML 為基礎，這使得它能同 HTML、CSS、DOM、XSL、JavaScript、CGI 一樣成為一種新的標準。

2. 由純文字構成的圖形

SVG 是一種純文字構成檔案，經過瀏覽器解析後產生圖形。也就是說，可以不用任何影像處理工具，僅僅用文字編輯器就可以產生一個 SVG 圖形。

3. 向量圖形

SVG 向量圖形一般可縮放 16 次，至少四次，只有直線和方形的話可縮放 20 次以上，圖形物件是由線框和填充物所組成，由電腦根據向量資料進行計算，然後繪製而成，不會因為縮放或解析度問題而有點陣圖形的鋸齒狀。因此向量圖形相對於點陣圖有以下特色：

- (1) 文件的大小與圖形的純文字內容有關，而與圖形的具體尺寸無關。
- (2) 圖形的顯示大小可以放大、縮小，變化後不影響圖形檔案內容影響，所以在圖形檔內容文字不多的情況下，向量圖形具有儲存文字量小，又可縮放的優點。

由於向量圖形的這些特色，使得它尤其適合網路傳輸，網路向量圖形使用最多的是 Macromedia 公司的 SWF(Shockwave Flash)格式。但是這不是標準格式，而 SVG 的出現可能改變這種狀況。2005 年四月 Adobe 公司併購 Macromedia 公司，也許不久的未來兩者優點會結合在一起。

4. 開放式標準 (Open Standard)

SVG 之所以會出現是為了在網路上制定一種新的、大家認同的、可擴展的、

開放的圖形格式。這個標準是由許多的廠商和 W3C 所共同制定的，所以 SVG 就是一個開放性的產業標準，協助制定規格的廠商包括了：Adobe Systems、AOL、蘋果電腦、AutoDesk、Bitflash、Cannon、CSIRO(Australia's Commonwealth Scientific & Industrial Research Organization)、Eastman Kodak, Ericsson, Excrosoft, Hewlett-Packard, IBM, ILOG, IntraNet Systems, KDDI, Macromedia, Microsoft, Nokia, OASIS, Openwave, Opera, Oxford Brookes University, Quark, Savage Software, Schemasoft, Sun Microsystems, Xerox and ZoomOn。所參與的單位幾乎涵蓋了全球主要的軟體、系統供應商。可以想像的是這項技術的影響層面，就是一統天下所有平台的影像動畫規格。

5. 多種物件格式靈活運用的文件

SVG 檔案由三部分組成：向量圖形、文字和影像，可以直接使用影像，也可以產生向量圖形和文字物件，還可以引入影像而製作出任何其他格式圖形所能達到的效果。由於文件格式是純文字形式，因此可以很容易重新修改，而且其中的圖形文字描述還可以在一個圖檔內重覆使用。

6. 可提供圖形檢索

一般點陣圖，只能憑檔名在網際網路上被瀏覽器、搜索引擎檢索，同一圖形可能重複命名，而相同命名卻又內容不同的圖形，所以網路上搜尋文字容易，而搜尋適用圖片卻難，而 SVG 內嵌文字可供精確搜尋，此為 Flash 圖檔所不及。

7. SMIL 支援

製作網頁用的互動按鈕，其圖片和程式在以前是分開的，按鈕僅僅是一個圖片，按鈕的程式是由網頁的 Script 來作。SVG 改變這個規則，它經由 SMIL (Synchronized Multimedia Integration Language)，使得在圖片內時序元件可

控制，也就是說 SVG 支援了互動性的網頁效果。

8. 內嵌字元資料

製作網頁的技術在 DHTML (Dynamic Hypertext Markup Language) 中，可使用系統中沒有的字體，該字體在瀏覽時，可透過 Web 即時下載，但是這對英文系統來說是適合的，因為英文字體的檔案大小一般在幾十 KB 左右，而對於中文系統來說，這種方法不適合，因為任何一種中文字形的檔案大小都在 2MB 以上甚至超過 5 MB，一般使用者不可能為了觀看幾個中文字而浪費大量時間去等待。SVG 採用了一種動態字體的方法，它嵌入了圖形中所出現的文字字型資料，用戶不需要下載所有的字形，這對於中文使用者來說是相當便利的，尤其用在圖書館典藏系統，當 BIG5 字碼不敷使用時，有新的方法可以解決中文問題。

2.5.3 SVG 的常用語法

1. 基本用法

(1) 起始宣告和 XML 相同，第二行是 SVG 標準的宣告，一個簡單的輪廓如下：

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
"http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg>
    所有程式碼在此，注意 tag 要上下對稱
<!-- 這是註解 -->
</svg>
```

(2) 就像 XML 一樣使用巢狀結構，element 是有區分大小寫的，一般來說 SVG 用的 element 小寫居多，若有一個字母名稱大小寫錯誤就不行：

```
<clipPath ...>
```

<!-- 內文可以放這裡 -->

</clipPath>

<ClipPath ...>

<!-- 有一個字母名稱大小寫錯誤，內文無法放這裡 -->

</ClipPath>

(3) 傳遞 Attribute 單引號雙引號都可以，但不可以混用如"0'：

<rect x="0px" y='0px'/>

(4) 座標的起始通常指左上角座標，長度可使用的單位：

em ---- font size

ex ---- The x-height of the current font-size

px ---- Pixels (省略單位的話，以此 px 為預設值)

pt ---- Points (" There are 72 points to 1 inch ")

cm ---- Centimeters.

mm ---- Millimeters.

pc ---- Picas. There are 6 picas to 1 inch.

% ---- 瀏覽器 windows 百分比

in ---- 英吋

(5) 顏色的表示方法

語法： <style="stroke:#顏色 ;fill:#填充色 ;stroke-width:寬度 "/>

a. 用十六進位數值表示顏色

#FF0000 表紅色，若字母相同，可縮寫成#F00【例如#CCFFCC 寫成#CFC】

b. 用 rgb(0~255,0~255,0~255) 表示顏色

【例如 rgb(255,0,0)表紅色，還可以用百分比 rgb(100%,0%,0%)】

c. 用英文代碼 blue , red , black

表 2-1 常用顏色對照表

顏色	十六進位數值	rgb 值	英文代碼
黑色 ◆	#000000	(0 , 0 , 0)	black
白色 ◆	#FFFFFF	(255,255,255)	white
灰色 ◆	#808080	(128,128,128)	gray
銀色 ◆	#C0C0C0	(192,192,192)	silver
金色 ⦿	#FFD700	(255,128,128)	glod
紅色 ◆	#FF0000	(255, 0 , 0)	red
粉紅 ◆	#FFC8CB	(255,200,203)	pink
洋紅 ⦿	#FF00FF	(255, 0 ,255)	fuchsia
橙色 ⦿	#FFA500	(255, 165, 0)	orange
黃色 ⦿	#FFFF00	(255, 255, 0)	yellow
橄欖 ◆	#808000	(128, 128, 0)	olive
綠色 ⦿	#008000	(0 ,128, 0)	green
亮綠 ⦿	#00FF00	(0 ,255, 0)	lime
藍色 ⦿	#0000FF	(0 , 0 ,255)	blue
青色 ⦿	#00FFFF	(0 ,255,255)	cyan
海藍 ⦿	#000080	(0 , 0 ,128)	navy
藍綠 ⦿	#008080	(128,128, 0)	teal
紫色 ⦿	#800080	(128, 0 ,128)	purple
褐色 ⦿	#A52A2A	(165, 42 ,42)	brown

2.SVG 常用的圖形語法

(1)線段 line

語法： `<line x1="起點 x 座標" y1="起點 y 座標" x2="終點 x 座標",y2="終點 x 座標" stroke="顏色"/>`

範例： `<line x1="50px" y1="50px" x2="350px" y2="200px" stroke="red"/>`

說明： 這是一條紅色線段，起始點(50,50)，終點(350,200)，任何圖形若沒指定寬度，預設值為 1，圖形如左下。

(2)矩形 rect

語法： `<rect x="左上 x 座標" y="左上 y 座標" width="寬度", height="高度" style="stroke:#顏色 ; fill:#顏色"/>`

範例： `<rect x="450" y="100" width="150" height="120" style="stroke:#000000; fill:#FFFF00"/>`

說明： 矩形左上座標(450,100)，寬 150 pixels，高 120 pixels，圖形外框線條是黑色，內部填充是黃色的，圖形如右下。



圖 2 - 7 SVG 線段與矩形

(3)圓形 circle

語法： `<circle cx="圓心 x 座標" cy="圓心 y 座標" r="半徑" style="stroke:#顏色 ; fill:#顏色 ; stroke-width:寬度 "/>`

範例： `<circle cx="180" cy="140" r="3cm" style="stroke:rgb(128, 0 ,128); fill:#CCCCCCFF; stroke-width:2 " />`

說明： 這是一個圓形，圓心座標(180,140)，半徑 3 公分，外框線條是紫色 `rgb(128, 0 ,128)`，內部填充是灰色，`stroke-width:2` 表示寬度為 2 pixels，線段與矩形也可以用 `stroke-width:2` 這樣設定寬度。

(4)橢圓形 ellipse

語法： `<ellipse cx="圓心 x 座標" cy="圓心 y 座標" rx="x 軸半徑" ry="y 軸半徑" style="stroke:#顏色 ; fill:#顏色 ; stroke-width:寬度" />`

範例： `<ellipse cx="520" cy="140" rx="120" ry="60" style="stroke:#FF0000; fill:#00FFFF; stroke-width:3" />`

說明： 這是一個橢圓形，圓心座標(520,140)，x 軸半徑 120，y 軸半徑 60，外框線條是紅色#FF0000，寬度為 3pixels，內部填充是青色的。

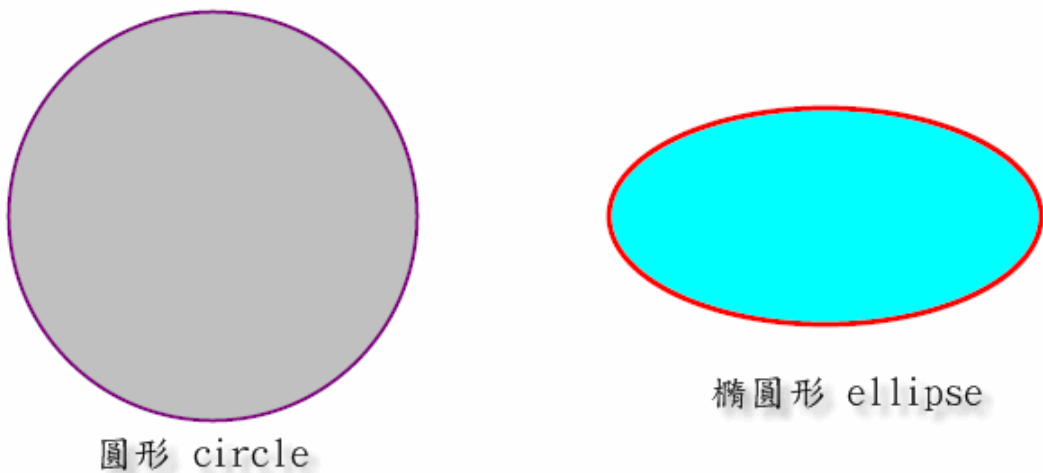


圖 2 - 8 SVG 圓形與橢圓

(5)多邊線形 polyline

語法： `<polyline points="點座標一 點座標二 點座標三" style="stroke:#顏色 ; fill:#顏色 ; stroke-width:寬度" />`

範例： `<polyline style="stroke:#000000; fill:# 00FFFF; stroke-width:2" points="60,200 60,50 160,50 230,140"/>`

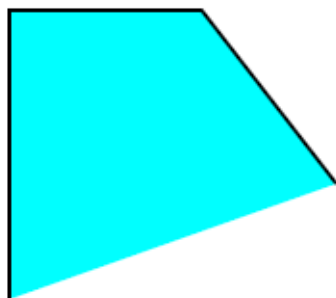
說明： 多邊線形圖形如左下，起始座標(60,200)，點座標二(60,50)，點座標三(160,50)，點座標四(230,140)，形成一個有缺口的四邊形，外框線段顏色為黑色，寬度為 2，內部填充是青色的。

(6)多邊形 polygon

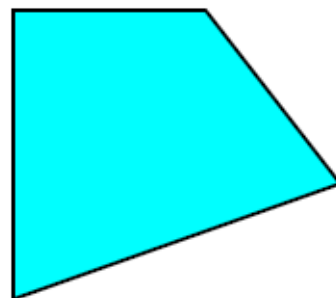
語法：`<polygon points="點座標一 點座標二 點座標三" style="stroke:#顏色 ;fill:#顏色 ;stroke-width:寬度 ">`

範例：`<polygon style="stroke:#000000; fill:# 00FFFF; stroke-width:2" points="360,200 360,50 460,50 530,140 360,200"/>`

說明：多邊線形圖形如左下，起始座標(360,200)，點座標二(360,50)，點座標三(460,50)，點座標四(530,140)，自動連回起始點，形成一個有封閉的四邊形，外框線段顏色為黑色，寬度為2，內部填充是青色的，若 polyline 頭尾座標點重疊就可以形成封閉區域，將與 polygon 有相同結果。



多邊線形 polyline



多邊形 polygon

圖 2 - 9 SVG 多邊線形與多邊形

3.SVG 常用的路徑 path

path 中配合屬性的常用 method 有下列 5 種：

- moveto --- Moves the pen to a particular point, while it is raised above the rendering (用 M 表示)

- lineto --- Draws a straight line to a defined point (用 L 表示)
- curveto --- Draws a Bezier curve to a define point (用 C 表示)
- arc --- Draws an elliptical or circular arc to a defined point (用 A 表示)
- closepath --- Closes a path (用小寫 z 表示)

以上字母 M (表示 moveto)、L (表示 lineto)、C (表示 curveto)、A (表示 arc) 在 path 中是絕對座標，如果小寫 m、l、c、a 就是相對座標。

(1)直線路徑

語法： `<path attribute="method1 座標 x1 座標 y1, method2 座標 x2 座標 y2, 座標 x3 座標 y3,method3 />`

範例： `<path d=" M 50 50 , L200 50 , 200 200 , 200 200 , z " style="fill:none ; stroke:red; stroke-width:5"/>`

說明： **d** 是最常用的 attribute，**M 50 50** 表示先移動到起始點(50,50)，**L200 50** 表示畫直線到(200,50)，座標 **200 200** 表示繼續剛才的畫直線方法，繼續畫到(200,200)，最後以小寫 **z** 結尾，表示回到起始點，完成一個封閉路徑，**stroke:red** 表示紅色路徑，**stroke-width:5** 表示寬度 5。此範例圖形呈現結果和 `<rect x="50px" y="50px" width="150px" height="150px"` 直接畫矩形是一樣的，但大量使用 PATH 後，可使程式碼縮減，檔案變小。

當兩個連續的 method 是相同時：(省略第二個) 例如：L200 50 , L200 200 寫成 L200 50 200 200，範例中 **【path d="M50 50,L200 50, 200 200 , 50 200, z"】** 把它解釋成：拿起筆到(50,50)，畫一直線到(200,50)，再畫一直線到(200,200)，再畫一直線到(50,200)，關閉並畫直線回到起始點。其實還可以再省略逗點符號成為

【`path d="M50 50 L200 50 200 200 50 200 z"`】，如果連指令 L 前的空白都省略，變成【`path d="M50 50L200 50 200 200 50 200 z"`】，但是這樣閱讀困難，除了第一個指令 M，其他 L 可以省略變成【`path d="M50 50 200 50 200 200 50 200 z"`】。

水平和垂直的 lineto 指令 (H and V)【`path d="M 50 50 , L 200 50 ,200 200 , 50 200 ,z"`】可以寫成【`path d="M 50 50 ,H 200 ,V 200 ,H 50 ,z"`】，你要是喜歡的話擠成一團也可以【`path d="M50 50H200V200H50z"`】。

使用小寫字母的 method 表示相對座標，【`path d="M50 50,l 150 0, 0 150 , -150 0, z,m 50 50,l 50 0, 0 50, -50 0, z"`】，解釋成：拿起筆到(50,50)，畫一直線往右 150、上下不動，再畫一直線水平不動、往下 150，再畫一直線往左 150、上下不動，關閉並畫直線回到起始點。此時畫出的矩形和【`rect x="50px" y="50px" width="150px" height="150px"`】直接畫矩形是一樣的。接下來在內部畫一個更小的矩形，整個圖形看起來像是一個「回」字，如下圖。

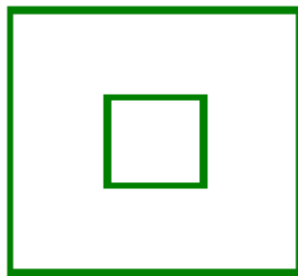


圖 2 - 10 SVG 的 path 路徑圖

(2) 曲線路徑

SVG 的 path 提供三種曲線產生方式：Cubic Bezier curve、Quadratic Bezier curve、Arc Bezier curve，能夠直接繪製曲線，還可以在曲線上走文字。下表為曲線 method 在 path 中的簡寫字母代碼：

表 2 - 2 path 中曲線簡寫字母代碼表

Attribute	絕對／相對座標	shorthand syntax
Cubic Bezier curve	C/c	S/s
Quadratic Bezier curves	Q/q	T/t
Elliptical arc curves	A/a	

語法： <path attribute="method 起始座標 x0 y0 Curve 種類 控制點 x1 y1
x2 y2 結束點 x y />

範例一：<path d=" M 90.7895 235.526 C 7.91503 121.776 30.9618 38.2115 226.316
207.895" style="fill:none ; stroke:rgb(0,0,0); stroke-width;4"/>

說明： M 90.7895 235.526 表示先移動到起始點(90.7895 235.526)，C 為
Cubic Bezier curve 的簡寫字母代碼，控制點座標有四個 7.91503
30.9618 38.2115 ，最後兩個是結束點 226.316 207.895 ，
stroke:rgb(0,0,0); stroke-width;4 表黑色線條寬度 4，圖形如左下。



圖 2 - 11 SVG 的 Cubic Bezier 曲線路徑圖

範例二： <path d=" M 96.0526 151.316
C102.973 102.874 229.734 24.9198 214.474 122.368
C207.901 164.344 191.176 238.14 227.632 226.316
C284.063 208.012 375.766 35.3111 367.105 203.947"
style="fill:none ; stroke:rgb(0,0,0); stroke-width;4"/>

說明： 起始點(96.0526 151.316)之後，共有三段連續 Cubic Bezier 曲線，
每一段均有兩個控制點，和一個結束點。

4.文字 Text

(1)基本用法

語法： `<text x="x 座標" y="y 座標" style="font-size: 點數; font-family: 字型; fill: 顏色"> 文字字元 </text>`

範例： `<text x="250" y="150" style="font-size: 40; font-family: Arial; fill: red"> NTNU </text>`

說明： 文字起始座標(250,150)，40 點字 Arial 字型，紅色 NTNU 四個字。

文字字元也可以用 ASCII 碼表示，【N T N U】相當於 NTNU 四個字，所以範例也可寫成`<text x="250" y="150" style="font-size: 40; font-family: Arial; fill: red">N T N U </text>`。Font-family 是 CSS 2 語法，這種方法可以顯示超過 128 以上的不可見字碼。例如 Arial 字型代表 copy right 符號©，® 代表註冊符號®，但換成 Sanmin Special 字型代表©符號，® 代表®符號。

(2)中文字體顯示

要在 Adobe SVG Viewer 當中顯示中文字體，需要將中文字體名稱進行“國際化”，先要把檔頭宣告 encoding="UTF-8"，還要把 svg 檔本身存成 utf 編碼形式！並使用正確的中文字體即可顯示中文。SVG 支援 US-ASCII、ISO-8859-1、UTF-16 和 UTF-8 四種編碼方式，使用亞洲雙位元語系如繁體中文、韓文、日文都可以。

語法： `<text x="x 座標" y="y 座標" style="font-size: 點數; font-family: 字型; fill: 顏色"> 文字字元 </text>`

範例： `<text x="250" y="150" style = "font-size : 40; font-family : 'PMingLiU'; fill : red">師大資訊教育</ text >`

說明： 文字起始座標(250,150)，大小為 40 點字，'PMingLiU'是 Windows 上的'新細明體'，如下圖呈現紅色「師大資訊教育」四個字。



圖 2 - 12 用 SVG 顯示中文字

在 SVG 中直接嵌入中文字體，便可以讓其他人都以你所嵌入的字體，看到你在設計時所使用的字體效果，而他們的電腦裡並不需要裝這個字體，只要將中文字體名稱改為英文名稱，如將“宋體”改為" Simsun "，“黑體”改為" Simhei "，“仿宋”改為" FangSong "，“楷體”改為" KaiTi "，“隸書”改為" LiSu "。

5.位移與旋轉

在 SVG 繪圖的座標系統裡，原點在螢幕左上方，往右為 x 軸正向，往下為 Y 軸正向，圖形中任何物件：shape、text、path.....，只要有用到參考座標的，就可以把物件位移或旋轉，最常在 g 元素下使用 transform 屬性，主要 method 如下：

- translate(x,y)：水平移動 x，垂直移動 y。
- rotate(degree,Posx,Posy)：旋轉 degree，旋轉軸心座標 x、y。
- scale(x rate, y rate)：縮放 x 軸、y 軸分量 rate 倍數。
- skewX(degree)：水平傾斜 degree。

- skewY(degree) : 垂直傾斜 degree。
- matrix(a,b,c,d,e,f) : 矩陣運算

下圖將虛線的 L 形垂直線段經過四次旋轉指令：`transform="rotate(30)"`、`transform="rotate(30, 200,0)"`、`transform="rotate(30, 400,0)"`、`transform="rotate(30, 600,0)"` 由下而上依次改做成四條 L 形實線，而虛線和實線的四個交點座標：`(0,0)`、`(200,0)`、`(400,0)`、`(600,0)`正好是旋轉軸心座標。

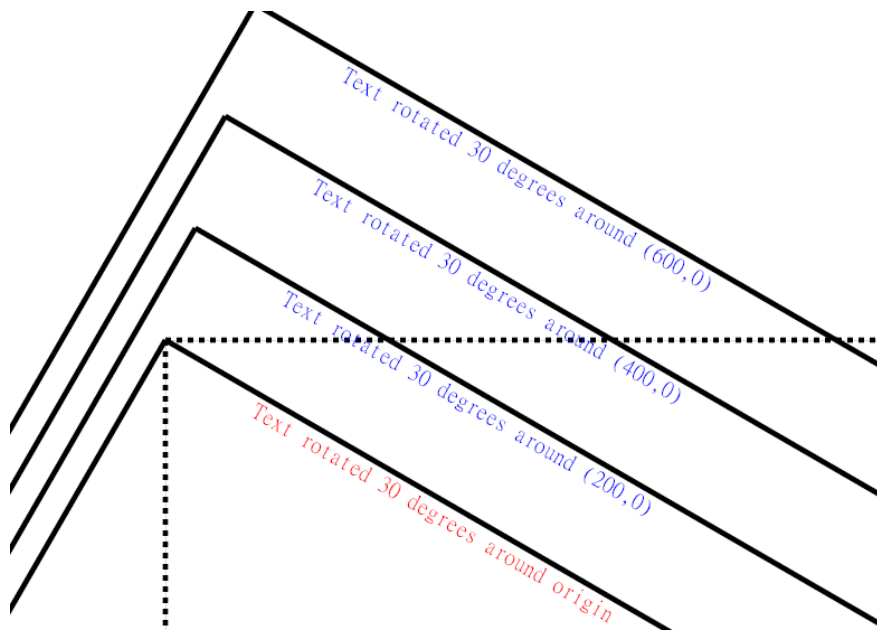


圖 2 - 13 SVG 的直線、文字旋轉圖

矩陣運算與旋轉、平移的關係如下：

- 平移量 tx、ty：指令 `translate(tx,ty)=matrix(1,0,0,1,tx,ty)`。
- 旋轉 θ 度：指令 `rotate(θ)=matrix(cos θ , sin θ , -sin θ , cos θ , 0, 0)`。
- x、y 軸縮放 sx、sy 倍：指令 `scale(sx, sy)=matrix(sx,0,0,sy,1,1)`。
- 水平傾斜 θ 度：指令 `skewX(θ)=matrix(1,0,tan θ ,1,0,0)`。
- 垂直傾斜 θ 度：指令 `skewY(θ)=matrix(1,tan θ ,0,1,0,0)`。

6. 動畫 animate

SVG 文件動畫呈現，以時間軸為依據，指定目標元件後，目標元件的屬性值就根據時間的變化而產生動畫。時間相關的屬性有：begin、dur(how long)、restart、repeatCount(specified number of times)、fill；影響動畫的屬性有：from、to、by、calcMode(discrete、linear、paced、spline)、values、keyTimes(control the speed)。

在 SVG 中使用動畫，遵循了 SMIL 2.0(Synchronized Multimedia Integration Language)規範，主要動畫元素如下：

- Set：是 animate 的簡化，主要用來改變非數值屬性的值，例如透明度。
- Animate：改變數值屬性的值，例如時間、座標。
- animateColor：改變某些元素的顏色屬性值。
- animateMotion：沿著 path 移動的 SVG 元素。
- animateTransform：使用 transform 改變屬性值。

下面的 SVG 範例程式碼，是使用 animateMotion 元素來產生動畫效果。

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<svg width="" height="" >
<path d="M300 300 l200 0 0 200 -200 0 z"
style="fill:none; stroke:black; stroke-width:4" />
<g style="fill:none; stroke:red; stroke-width:7;" transform="translate(200,200)">
  <animateMotion path="M 0 0, 1 200 0, 0 200, -200 0, 0 -200"
  begin="2s" dur="8s" repeatCount="indefinite" rotate="auto"/>
  <line x1="0" y1="0" x2="50" y2="0" />
  <line x1="50" y1="1" x2="35" y2="-15" />
  <line x1="50" y1="-1" x2="35" y2="15" />
</g>
</svg>
```


上面例子中，我們可以看到，一個紅色箭頭，繞著一個正方形移動旋轉，屬性 `begin="2s" dur="8s"` 表示動畫從正方形出現後的第 2 秒才開始，時間到第八秒結束，共持續了 6 秒，平均每一個邊停留 1.5 秒，`repeatCount="indefinite"` 則是無限回圈，所以紅色箭頭會不段的繞，以上程式的效果如下圖所示：

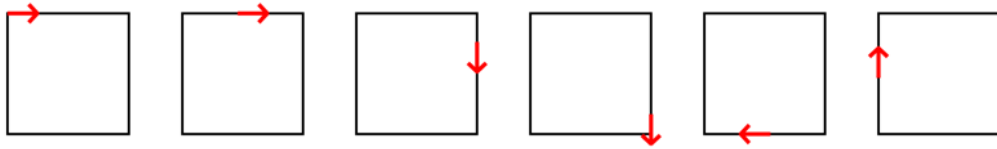


圖 2 - 14 SVG 的動畫分解圖