



13-5 XML 簡介

- ➡ XML文件結構
- ➡ 文件物件模型
- ➡ 文件型態定義
- ➡ Xpath標準





13-5 XML 簡介

- **全球資訊網**(World-Wide-Web ; 簡稱WWW或Web)已經可以說是全世界資訊分享的主要方式，因為它在無遠弗屆的廣域網路環境中，提供一種便利且簡單的方式去存取資料，所以相當多的企業已經將其產品廣告或可分享的資料放在網際網路上。





13-5 XML 簡介

- ➡ WWW 上所接受的資料型態是符合 HTML 格式的文件，但是原本 HTML 是設計為顯示資料之用，以便將文件內容呈現在使用者面前，所以內含許多控制輸出的**標籤 (tag)**，如 `<table>`、``、``、``...等，而不是表示資訊的內容及它的結構，其缺乏對資訊意涵的描述，所以不利於自動化的資訊傳遞與交流。



13-5 XML 簡介

- ➡ 可延伸式標記語言 (Extensible Markup Language ; XML) , 已成為最近 Web 上相當受到重視的格式。
- ➡ XML 是由 W3C 制定的一個有關於描述資訊的上層語言(meta language) , 其 1.0 版於 1998 年 2 月正式推出。
- ➡ XML 的目的為定義一個描述資料之標準 , 允許使用者可以自由地定義標籤 , 以適當的結構來描述所要傳輸的資料。(參考鏈結)



XML (Extensible Markup Language)

- ▶ 類似 HTML，利用標籤來定義資料。
- ▶ XML 著重在如何描述文件內容，而 HTML 重點在如何展示一份文件。

```
<table border=2 bordercolor=black>
```

```
<tr>
```

```
<th> 姓名 </th>
```

```
<th> 國文 </th>
```

```
<th> 數學 </th>
```

```
</tr>
```

```
<tr>
```

```
<td> 小明 </td>
```

```
<td> 90 </td>
```

```
<td> 76 </td>
```

```
</tr>
```

```
<tr>
```

```
<td> 小華 </td>
```

```
<td> 85 </td>
```

```
<td> 81 </td>
```

```
</tr>
```

HTML

姓名	國文	數學
小明	90	76
小華	85	81

```
<?xml version="1.0" encoding="Big5"?>
```

```
<成績記錄>
```

```
<學生>
```

```
<姓名> 小明 </姓名>
```

```
<國文> 90 </國文>
```

```
<數學> 76 </數學>
```

```
</學生>
```

```
<學生>
```

```
<姓名> 小華 </姓名>
```

```
<國文> 85 </國文>
```

```
<數學> 81 </數學>
```

```
</學生>
```

```
</成績記錄>
```

XML





13-5 XML 簡介

- ➡ XML 規格將資料與使用者介面分離，所以易於達到自動化處理的目的。
- ➡ XML 對於各類型資料(如物件、文章、圖形、文字檔、二元檔等)都能標註，且以文字為基礎來表示資料，不僅容易在異質系統之間傳遞交流，且能穿過防火牆，便於在不同企業間進行資料交換。
- ➡ 所以，已經有相當多組織，將其資料表示標準化，以 XML 格式表示，作為在 Web 上資料共享的主要依據。



13-5 XML 簡介

- ➡ 從 XML 提出至今，幾家重要的軟體廠商，如 Oracle、HP、IBM、Microsoft 等，皆全力的投入研發，並配合 W3C 的規格，不斷地推出新的產品與應用，所以 XML 的重要性絕對是不可忽視。
- ➡ 由於 XML 只提供表示資料的方式，必須配合 W3C 另外定義的輔助技術來處理 XML 資料。
- ➡ XML 的文件可給人與機器閱讀
 - ▶ 在物聯網 IoT 的環境中大量應用。





XML 文件結構

- ➡ 一個 XML 文件的範例如圖 13-18 所示，它表示了三本書籍的資料，每一本書則分別描述了書名、作者、出版廠商、出版日期等訊息。
- ➡ 由範例中，可看出 XML 文件利用適當的標註，來提供資料的結構及與語意有關的資訊。
- ➡ 特別注意的是：該文件只表示「資料」，並無指定「顯示介面」。



XML 文件結構

```
L1 <Books amount="3">
L2 <Book>
L3   <Title>Essential XML</Title>
L4   <Authors>
L5     <Author>Box</Author>
L6     <Author>Skonnard</Author>
L7     <Author>Lam</Author>
L8   </Authors>
L9   <Publisher>AW</Publisher>
L10  <Date year="2000" month="7"/>
L11 </Book>
L12 <Book>
L13   <Title>計算機概論 </Title>
L14   <Authors>
L15     <Author>趙坤茂</Author >
L16     <Author >張雅惠</Author>
L17     <Author >黃寶瑩</Author >
L18   </Authors>
L19   <Publisher>全華</Publisher>
L20   <Date year="2004" month="7"/>
L21 </Book>
L22 <Book>
L23   <Title>Spanning Trees and Optimization Problems</Title>
L24   <Authors>
L25     <Author>吳邦一</Author>
L26     <Author>趙坤茂</Author>
L27   </Authors>
L28   <Publisher>Chapman & Hall/CRC Press, USA. </Publisher>
L29   <Date year=" 2004" month="1"/>
L30 </Book>
L31 </Books>
```

XML文件範例





XML 文件結構

- ➡ 以下利用此範例進一步說明 XML 文件的架構。XML 可說是由一個個**元素(element)**所組成的。
- ➡ 所謂的元素，就是由一個**開始標籤(start-tag)**到對應的**結束標籤(end-tag)**為止，包含其中的所有內容。
- ➡ 譬如，從 L2 行的開始標籤 `<Book>` 到 L11 行的 `</Book>` 為止，表示了一個 Book 元素。



XML 文件結構

- ➡ 元素中可以包含其他元素，稱作子元素，譬如，該 Book 元素有四個子元素：Title、Authors、Publisher、Date。
- ➡ XML 要求文件必須格式正確(well-formed)，也就是每個 XML 文件中只能有一個在最外層的根元素(root element)，如 L1-L31 行的 Books 元素；同時，每個元素的開始標籤與結束標籤須成對，標籤之間不可交錯，即所有元素的排列必須為嚴謹的巢狀結構。



XML 文件結構

- ➡ 元素可包含屬性(attribute)，所有的屬性之值必須加上單引號或雙引號，如 L10 行的 Date 元素包含一個屬性 year，其值為 2000。





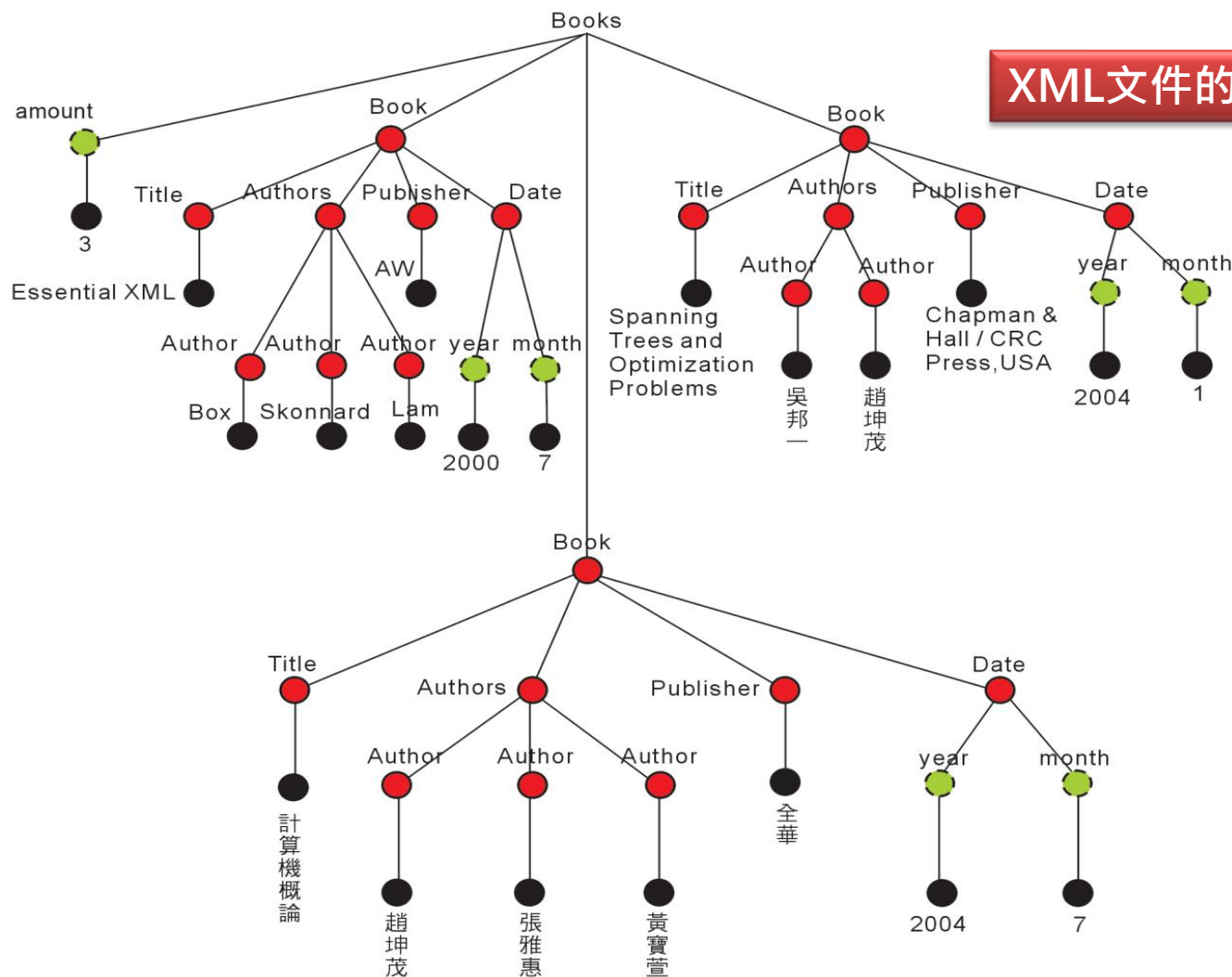
文件物件模型

- ➡ 文件物件模型 (Document Object Model ; DOM) , 是 W3C 定義來描述 XML 文件的架構 , 同時規範存取 XML 資料的介面 , 然後各家廠商可以根據該標準介面 , 自行提供實作的細節。
- ➡ DOM 的基本觀念 , 就是將 XML 檔案分解成個別的元素、屬性等 , 然後以它們為節點 , 表示成一個有順序的標籤樹(ordered label tree)。



文件物件模型

XML文件的DOM樹狀表示





文件型態定義

- ➡ 相較於結構化的關聯式資料庫，XML 提供了一種半結構化的(semi-structured)表示資料的方式。
 - ▶ 舉例來說，關聯式資料庫的每筆資料都必須具有固定個數的欄位，但是 XML 資料內某一個元素可以出現一次以上。
- ➡ 這些資料定義規格，可以利用 W3C 所頒訂的**文件型態定義**(Document Type Definition ; DTD)來描述。





文件型態定義

- ▶ 該定義可規範特定 XML 文件的格式，也就是將每一個元素可以包含哪些子元素或屬性、各元素出現的順序等，清楚地加以定義和規範。

L1	<?xml version="1.0" encoding="Big5"?>
L2	<!ELEMENT Books (Book*)>
L3	<!ATTLIST Books amount NMTOKEN "1">
L4	<!ELEMENT Book (Title, Authors, Publisher, Date)>
L5	<!ELEMENT Title (#PCDATA)>
L6	<!ELEMENT Authors (Author*)>
L7	<!ELEMENT Publisher (#PCDATA)>
L8	<!ELEMENT Date EMPTY>
L9	<!ATTLIST Date year #CDATA #REQUIRED month #CDATA #REQUIRED>
L10	<!ELEMENT Author (#PCDATA)>

DTD範例





文件型態定義

- ➡ 在 DTD 文件中，ELEMENT 標籤之後定義的是元素的名稱，接著用小括號括起來的，是該元素的「內容模型」，也就是在對應的 XML 文件中可以出現的內容。
- ➡ 在 L4 行的定義，規定了 Book 這個元素可包含 Title、Authors 等四個子元素。
- ➡ 至於 L3 行的「NMTOKEN」則是規範屬性值內不能包含空白。



文件型態定義

- ➡ 在 L5 行中進一步註明Title元素存放的資料為「#PCDATA」(Parsable Character Data)，該字串為預先定義的標記，代表可解析的文字資料。
- ➡ ATTLIST 標籤則是用以宣告元素的屬性，包含了屬性名稱、屬性類別及預設行為的描述，若屬性不只一個時，可以用這三個部分為一個單位一直重複下去。





文件型態定義

- ▶ L9 行定義了 Date 這個元素具有 **year** 和 **month** 這二個屬性。其中，**year** 屬性的類別為 **#CDATA**，表示該屬性值為一般文字；預設行為的描述為 **#REQUIRED**，表示該屬性值一定要存在。
- ▶ 特別注意的是：DTD 允許類似 **regular expression** 的符號，如 L2 行的星號(*)代表 **Books** 元素裡可包含多個 **Book** 子元素。



XPath 標準

- ➡ XPath 是節點位置語言，用來取出位於特定位置的 XML 元素。
- ➡ 我們直接參照圖 13-19 的樹狀結構，可以更清楚地瞭解 XPath 的寫法。
- ➡ 在 XPath 中，我們必須指定如何從一個節點走到下一個節點，最常見的是把從根元素到該節點的完整路徑寫出來，如「/Books/Book/Title」。



XPath 標準

- ➡ 該 XPath 敘述式，會回傳下列三個元素：

```
<Title>Essential XML</Title>  
<Title>計算機概論</Title>  
<Title>Spanning Trees and Optimization Problems</Title>
```

- ➡ 有時我們不是很確定元素在樹裡的確切位置，則我們可以在節點間使用「//」符號。
 - ▶ 在 XPath 的標準中，「/」代表元素間具有父子關係，而「//」則代表元素間具有祖孫關係。



XPath 標準

- ➡ 假設我們不確定 Title 元素的完整路徑，可以下「//Title」或「/Books//Title」，其回傳的元素會跟上式「/Books/Book/Title」相同。
- ➡ 可以利用萬用字元，取出不限定名稱的所有元素，如「/Books/Book/*」。
- ➡ 也可針對某個節點的內容加以限制。





XPath 標準

- ➡ 「/Books/Book[//Author= “趙坤茂”]/Title」，
會回傳所有「趙坤茂」寫的書本的標題。
- ➡ XPath 標準裡也提供許多函數以便使用者下達複雜的限制式，譬如表示式「//*[count(Book)=3]」，
則會指出擁有三個「Book」子元素的所有元素。

