

## Ch. 4 XML語法檢查DTD

- 什麼是DTD
- 在XML文件使用DTD
- DTD元素
- DTD屬性清單
- DTD實體宣告
- DTD記法宣告
- 編輯XML文件的DTD

## 什麼是DTD

- 什麼是Schema
- 什麼是DTD
- DTD建立資料交換的標準
- DTD與XML Schema

## 什麼是Schema

- Schema屬於一種XML文件內容的驗證機制，主要是定義文件出現的元素和屬性，它同時可以定義整份文件的架構，也就是元素擁有那些子元素的「內容模型」(Content Models)，內容模型定義元素合理的順序和巢狀的架構。
- XML 1.0版的Schema預設為DTD。

## 什麼是DTD

- 「DTD」(Document Type Definition)是XML 1.0版規格的一部分，它是XML文件的驗證機制，屬於XML文件組成的一部分，DTD在XML文件所扮演的角色就是定義XML文件的元素架構、元素標籤和屬性。
- 事實上XML 1.0版的規格大部分都是在處理DTD，如果使用者曾經使用過SGML，就會似曾相識的發現DTD的語法就是來自SGML DTD。

## DTD建立資料交換的標準

- XML文件提供應用程式一個資料交換的格式，DTD正是讓XML文件能夠成為資料交換的標準，因為不同的公司只需定義好標準的DTD，各公司都能夠依照DTD建立XML文件，並且進行驗證，如此就可以輕易建立標準一致和所需的資料交換標準資料。

## DTD與XML Schema

- 本章的內容都屬於XML文件的驗證機制，不過使用的語法完全不同，其差異如下所示：
  - DTD擁有自己獨特的語法和撰寫方式，它和XML文件的標籤架構完全不同。
  - XML Schema本身就是一份XML文件，使用標籤語法定義其它的XML文件，因為它就是一份XML文件，在學習上不必特別學習DTD的語法，而且支援資料型態，能夠定義元素的內容。

## 在XML文件使用DTD

- XML文件內部的DTD
- XML文件的外部DTD
- 混合內部和外部DTD
- 驗證 DTD 的 XML 文件

## XML文件內部的DTD

- XML文件能夠直接包含DTD，因為它本來就屬於XML文件的組成元素，DTD指令的位置是在XML文件開頭PI或註解的後面。  

```
<!DOCTYPE root_node [  
.....  
>
```
- DTD的語法是定義在<!DOCTYPE .. [...]>指令間，其參數的說明如下所示：
  - root\_node：XML文件的根標籤名稱。

## XML文件內部的DTD

```
01: <?xml version="1.0" encoding="Big5"?>
02: <!--網頁製作徹底研究系列-->
03: <!DOCTYPE booklist [
04: <!ELEMENT booklist (book+)>
05: <!ELEMENT book (code, title, authorlist, price)>
06: <!ATTLIST book
07:   sales (N | Y) #REQUIRED
08: >
09: <!ELEMENT code (#PCDATA)>
10: <!ELEMENT title (#PCDATA)>
11: <!ELEMENT authorlist (author+)>
12: <!ELEMENT author (#PCDATA)>
13: <!ATTLIST authorlist
14:   no CDATA #REQUIRED
15: >
16: <!ELEMENT price (#PCDATA)>
17: ]>
```

## XML文件內部的DTD

```
18: <booklist>
19:   <book sales="Y">
20:     <code>F8915</code>
21:     <title>ASP網頁製作徹底研究</title>
22:     <authorlist no="1">
23:       <author>陳會安</author>
24:     </authorlist>
25:     <price>580</price>
26:   </book>
27:   <book sales="N">
28:     <code>F8916</code>
29:     <title>ASP與IIS4/5網站架設徹底研究</title>
30:     <authorlist no="1">
31:       <author>陳會安</author>
32:     </authorlist>
33:     <price>550</price>
34:   </book>
35: </booklist>
```

## XML文件的外部DTD

- 上面的範例是將DTD語法直接插在XML文件，我們也可以將DTD的語法獨立儲存成一個檔案，然後在XML文件指定使用的DTD檔案。

```
<!DOCTYPE root_node SYSTEM "filename.dtd">
```

- 上述指令使用SYSTEM關鍵字指定XML文件使用的DTD，各參數的說明，如下所示：
  - root\_node：XML文件的根標籤。
  - filename.dtd：外部DTD檔案的路徑，通常使用的副檔名為.dtd。

## 混合內部和外部DTD

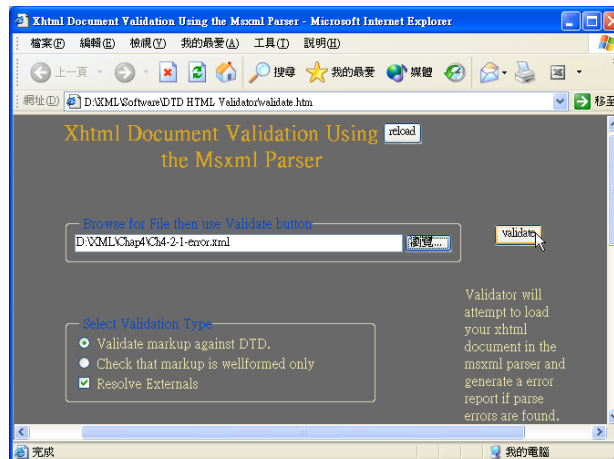
- XML文件可以同時使用內部和外部DTD，例如：我們可以使用現成的DTD，如果需要取代某些XML元素，就直接在XML文件加上DTD語法。

```
<!DOCTYPE root_node  
SYSTEM "filename.dtd"  
[  
  <!ELEMENT node (child_node+)>  
  .....  
>
```

- 上述指令除了加上外部DTD檔案外，也可以在「[」和「]>」間加上DTD語法。

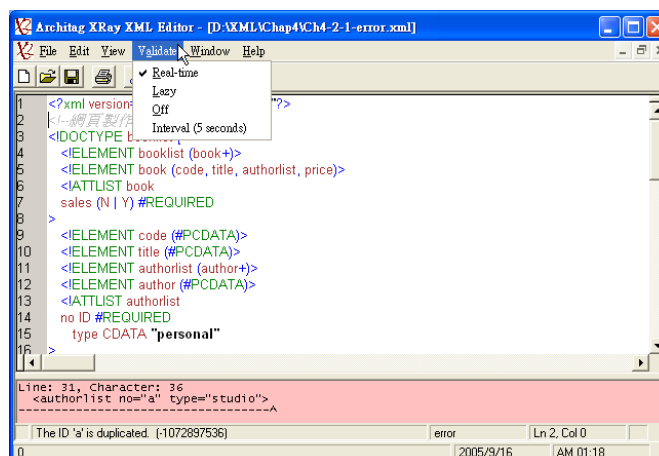
# 驗證DTD的XML文件

## Online Xhtml Validator



# 驗證DTD的XML文件-2

## XRay XML Editor的驗證方式



## DTD元素

- DTD元素的語法
- 定義XML文件的架構
- 定義XML元素的資料
- 定義混合內容(Mixed Content)

## DTD元素的語法

- DTD元素(Element)是宣告XML元素的語法，包含元素標籤、內含的子元素和元素內容資料，它同時也宣告XML文件的元素架構。

`<!ELEMENT element_name (element_or_data)>`

- DTD的XML元素宣告使用`<!ELEMENT .... >`指令，每個XML元素都需要宣告，其參數的說明如下：
  - element\_name：XML元素的標籤名稱。
  - element\_or\_data：定義元素內含的資料或是其下的子元素，如果不只一個請用「,」分隔。



## 定義XML文件的架構

- DTD元素的宣告不只在定義XML文件的標籤，它同時也定義元素間關係的XML文件架構，也就是元素是否擁有子元素。
- 如果元素擁有一到多個子元素，我們需要將子元素定義在括號中，例如：authorlist元素擁有子元素author，如下所示：  
`<!ELEMENT authorlist (author+)>`
- 如果元素擁有多個子元素，括號的子元素清單需要使用「,」分隔，例如：book元素擁有數個子元素，如下所示：  
`<!ELEMENT book (code, title, authorlist, price)>`

## 定義XML文件的架構

- 如果相同的子元素能夠重複，我們需要進一步的使用符號宣告各子元素出現的次數，如下表所示：

符號	說明
+	子元素最少出現一次
*	子元素出現零或多次
?	子元素出現零或一次
	子元素清單之一
(...)	將元素清單括起來

## 定義XML文件的架構

### 子元素只出現一次

- 如果元素名稱後沒有加上任何符號，表示子元素只能出現一次，如下所示：  
`<!ELEMENT book (code, title, authorlist, price)>`
- 上述book元素只會出現一個code、title、authorlist和price子元素，如果XML文件擁有兩個title元素就不符合DTD的定義。

## 定義XML文件的架構

### 子元素最少出現一次

- 如果子元素最少出現一次，而且可能出現很多次，此時請在元素後加「+」符號，如下所示：  
`<!ELEMENT booklist (book+)>`
- 上述booklist元素的子元素book最少一個。

## 定義XML文件的架構

### 子元素出現零或多次

- 如果子元素可能沒有或有很多個，此時請在元素後加上「\*」符號，如下所示：  
`<!ELEMENT note (message*)>`
- 上述note元素可能沒有子元素message，也可以有一或多個。

## 定義XML文件的架構

### 子元素出現零或一次

- 子元素是個選項，有或沒有，此時請在元素後加上「?」符號，如下所示：  
`<!ELEMENT note (message?)>`
- 上述note元素可能沒有子元素message，也可以只有一個。

## 定義XML文件的架構

### 子元素清單之一

- 如果子元素有多種選擇，可能為任何一個的多選一，此時各元素名稱請使用「|」符號分隔，如下所示：  
`<!ELEMENT note (message+, (table | image)*)>`
- 上述note元素的子元素message最少出現一次，或是table和image二選一，而且允許出現零或多個。

## 定義XML元素的資料

### 空元素

- 空元素表示在標籤間沒有任何資料，此時使請使用EMPTY關鍵字，通常這些標籤是指定相關的屬性，如下所示：  
`<!ELEMENT image (EMPTY)>`
- 上述image元素就是一個空元素，如下所示：  
`<image />`

## 定義XML元素的資料

### 元素資料

- 如果元素內擁有資料，此時的資料內容有二種，使用二個關鍵字#PCDATA和ANY，如下所示：

<!ELEMENT code (#PCDATA)>

<!ELEMENT message (ANY)>

- 上述message元素的資料內容。

## 定義混合內容(Mixed Content)

- 如果文字資料#PCDATA仍然包含有其它標籤，此時DTD定義的元素內容屬於一種混合內容，如下所示：

<!ELEMENT message (#PCDATA | quote | b)\*>

<!ELEMENT quote (#PCDATA)>

<!ELEMENT b (#PCDATA)>

- 上述message元素可以是文字資料，其中擁有quote子元素。

## DTD屬性清單

- DTD屬性的語法
- 屬性的宣告
- 屬性值的宣告
- XML的預設屬性

## DTD屬性的語法

- DTD屬性清單(Attribute List)宣告XML元素所擁有的屬性清單，包含屬性名稱和值，屬性的位置是放在開始標籤。

```
<!ATTLIST element_name attribute_name attribute_type  
default_value>
```

- XML元素的屬性都需要使用<!ATTLIST ....>宣告，可以同時宣告元素的多個屬性，使用空格分隔即可，其參數的說明如下：
  - element\_name：屬性所屬的XML元素名稱。
  - attribute\_name：屬性名稱。
  - attribute\_type：指定屬性值的種類。
  - default\_value：屬性的預設值，用來指出屬性是否需要出現或只是選項。

## 屬性的宣告

### 選項屬性(可有可無)

- 如果元素的屬性可有可無，並不強迫使用者一定需要加上屬性，我們需要使用#IMPLIED關鍵字，如下所示：  
`<!ATTLIST authorlist type CDATA #IMPLIED>`
- 上述DTD宣告authorlist元素的屬性type為選項，可有可無。

## 屬性的宣告

### 一定需要的屬性

- 如果元素的屬性一定要有，使用者一定不能忘了加上屬性，此時需要使用#REQUIRED關鍵字，如下所示：  
`<!ATTLIST authorlist no CDATA #REQUIRED>`
- 上述DTD宣告authorlist元素的屬性no為必需的屬性。

## 屬性值的宣告

### ID屬性值

- ID屬性值為唯一的值，如果在整份XML文件需要類似資料表的主索引，我們就可以將屬性設為ID屬性種類，如下所示：  
`<!ATTLIST authorlist no ID #REQUIRED>`
- 上述的屬性宣告指出authorlist元素的no屬性為必須的，而且擁有唯一的屬性值。

## 屬性值的宣告

### 屬性的預設值

- 如果XML文件沒有強迫設定元素的屬性，不過屬性需要一個預設值，以便應用程式處理時的需要，此時請指定屬性的預設值，如下所示：  
`<!ATTLIST authorlist type CDATA "personal">`
- 上述的屬性宣告指出authorlist元素的type屬性擁有預設值personal，XML元素如果沒有設定type屬性，屬性就使用此預設值。



## 屬性值的宣告

### 固定屬性值

- 如果需要固定值的屬性值，使用者並不能更改此屬性值，如下所示：

```
<!ATTLIST book company CDATA #FIXED  
"Microsoft">
```

- 上述的屬性宣告指出book元素的company屬性為固定值Microsoft，XML元素如果沒有設定company屬性，屬性就使用此固定值。

## 屬性值的宣告

### 屬性值清單

- 如果需要限制屬性值只能是其中幾個，例如：sales屬性值不是N就是Y，此時就是屬性值清單的選擇題，如下所示：

```
<!ATTLIST book sales (N | Y) #REQUIRED>
```

- 上述的屬性宣告指出sales元素的sales屬性只有二選一。

## XML的預設屬性-1

### 預設屬性xml:space

- 這個屬性設定空白是否有意義，以便正確的顯示文件的空白，事實上它只是一個開關，告訴應用程式需要注意空白字元。
- xml:space宣告的方式，如下所示：  

```
<!ATTLIST element_name xml:space (default | preserve)>
```
- 上述xml:space屬性的宣告，只有擁兩個值default和preserve。

## XML的預設屬性-2

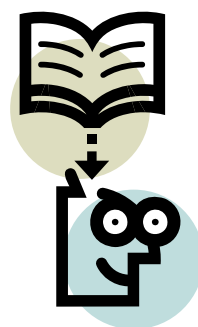
### 預設屬性xml:lang

- XML元素都可以指定xml:lang屬性，設定文件內容使用的語系，如下所示：  

```
<book xml:lang="zh-TW" sales="Y"> .... </book>
```
- 上述的xml:lang屬性指定book元素使用繁體字，如果為en表示為英文，la為拉丁文。

## 4-5 DTD實體宣告

- 內部實體
- 外部實體
- 參數實體



## 內部實體

- 「內部實體」(Internal Entities)指的是實體參考值屬於一個字串的文字內容。  
`<!ENTITY entity_name entity_value>`
- 上述的實體宣告使用`<!ENTITY .... >`宣告一個實體，其參數的說明如下：
  - entity\_name：實體名稱。
  - entity\_value：實體的內容，就是一個可以剖析的文字內容。
- 在XML文件使用實體的方式，稱為「實體參考」(Entity Reference)，以「&」開頭，「;」結尾，中間就是實體名稱，如下所示：  
`<book>&entity_name;</book>`

## 外部實體

- 「外部實體」(External Entities)對應的是外部的檔案，可以是文字或二進位檔案，文字檔案是插入文件，二進位檔案因為部析器無法處理，通常是作為屬性值，例如：圖片檔案。

<!ENTITY entity\_name SYSTEM url\_uri>

<!ENTITY entity\_name SYSTEM url\_uri NDATA notation\_name>

- 其參數的說明如下：
  - entity\_name：實體名稱。
  - url\_uri：實體參考檔案的位址路徑。
  - notation\_name：記法宣告的名稱，指定外部二進位檔案的種類。

## 參數實體

- 「參數實體」(Parameter Entities)定義的實體只能出現在DTD，主要的用途是在簡化DTD的語法。

<!ENTITY % entity\_name entity\_value>

- 實體名稱前的「%」表示此為參數實體，其參數的說明如下：

- entity\_name：實體名稱。
- entity\_value：實體的內容，通常為DTD語法或部分的宣告文字。

- 在DTD使用參數實體的方式是使用「%」開頭，「;」結尾，中間就是實體名稱，如下所示：

<!ELEMENT description (%content);>

## 4-6 DTD記法宣告

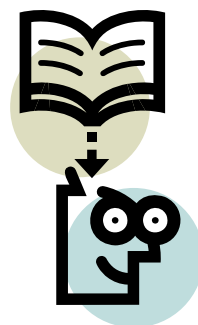
- 「記法宣告」(Notation Declaration)是用來宣告文件的資料是外部非XML的資料，特別是二進位的圖片檔案，以便應用程式知道如何處理這些資料。

```
<!NOTATION notation_name SYSTEM  
external_identifier>
```

- 上述的宣告使用<!NOTATION ....>宣告一個記法，其參數的說明如下：
  - notation\_name：記法宣告的名稱。
  - external\_identifier：外部資料的識別資料。

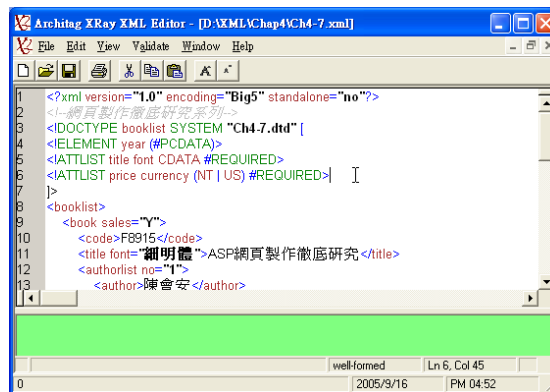
## 4-7 編輯XML文件的DTD

- 使用XRay XML Editor編輯DTD
- Schematron Validator驗證工具的使用
- 測試DTD的條件區塊



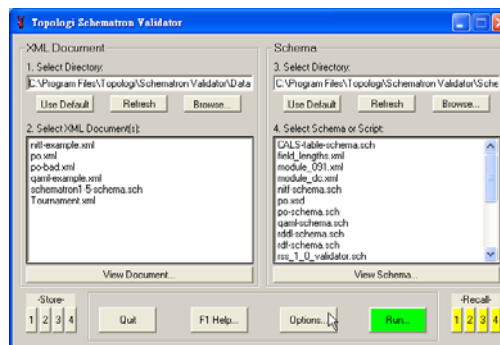
## 使用 XRay XML Editor 編輯 DTD

- DTD 宣告本來就是 XML 文件的一部分，我們可以在使用 XRay XML Editor 編輯 XML 文件時，在 XML 文件插入和編輯 DTD 宣告。



## Schematron Validator 驗證工具的使用

- Topologi Schematron Validator 是 Topologi P/L 公司開發的共享軟體 XML 驗證工具，支援 DTD、W3C XML Schemas 和 Schematron (另一種驗證語法) 驗證。



## 測試DTD的條件區塊-1

### IGNORE

<![IGNORE [dtd\_declarations]]>

- 上述的參數dtd\_declarations就是DTD的定義，如下所示：

<![IGNORE[<![ELEMENT code (#PCDATA)]]]>

- 上述的指令告訴XML剖析器忽略code元素，並不進行剖析。

## 測試DTD的條件區塊-2

### INCLUDE

<![INCLUDE [dtd\_declarations]]>

- 上述的參數dtd\_declarations就是DTD的定義，如下所示：

<![INCLUDE[<![ELEMENT code (#PCDATA)]]]>

- 上述的指令告訴XML剖析器需要處理code元素進行剖析。

## 作業

- 請為CH03的兩份XML文件撰寫驗證文件所需要的DTD (本題請存檔為(學號\_ch4\_1\_1.dtd及學號\_ch4\_1\_2.dtd)
- )